

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

**Інститут прикладного системного аналізу
Кафедра математичних методів системного аналізу**

«До захисту допущено»
В.О. завідувача кафедри
_____ О.Л. Тимощук

Дипломна робота

на здобуття ступеня бакалавра

з напрямку підготовки 6.050101 «Комп'ютерні науки»

**на тему: «Інформаційна система для прогнозування котирування акцій
на основі нейромереж»**

Виконав:

студент IV курсу, групи КА-55

Ночовний Олексій Олександрович _____

Керівник:

доц. кафедри ММСА,

доцент, д.т.н. Жиров О. Л. _____

Консультант з економічного розділу:

доцент, к.е.н. Шевчук О. А. _____

Консультант з нормоконтролю:

доцент, к.т.н. Коваленко А.Є. _____

Рецензент:

Зав. Каф. АУТС

НТУУ «КПІ ім. І. Сікорського»,

проф., д.т.н. Теленик С.Ф. _____

Засвідчую, що у цій дипломній роботі
немає запозичень з праць інших авторів
без відповідних посилань.

Студент _____

Київ – 2019 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Інститут прикладного системного аналізу
Кафедра математичних методів системного аналізу

Рівень вищої освіти – перший (бакалаврський)

Напрямок підготовки (програма професійного спрямування) – 120

Комп'ютерні науки (Системи штучного інтелекту)

ЗАТВЕРДЖУЮ

В.О.Завідувача кафедри

_____ О.Л. Тимошук

«__» _____ 20__ р.

ЗАВДАННЯ

на дипломну роботу студенту

Ночовному Олексію Олександровичу

1. Тема роботи «Інформаційна система для прогнозування котирування акцій на основі нейромереж», керівник роботи Жиров Олександр Леонідович, доцент, затверджені наказом по університету від «__»__ тберезня р. №__

2. Термін подання студентом роботи _____

3. Вихідні дані до роботи

4. Зміст роботи

5. Перелік ілюстративного матеріалу (із зазначенням плакатів, презентацій тощо)

Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Економічний	Шевчук О. А., доцент		

7. Дата видачі завдання _____

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка

Студент

О.О. Ночовний

Керівник роботи

О.Л. Жиров

РЕФЕРАТ

Дипломна робота: 76 с., 15 рис., 8 табл., 2 додатки, 19 джерел.

ЧАСОВІ РЯДИ, СТОХАСТИЧНІ МОДЕЛІ, СЕЗОННІСТЬ, ШТУЧНІ НЕЙРОННІ МЕРЕЖІ.

Моделювання та прогнозування часових рядів має принципове значення для різного практичного застосування. В зв'язку з цим, протягом останніх років у цій темі було безліч наукових робіт. У літературі запропоновано багато важливих моделей для підвищення точності та ефективності моделювання та прогнозування часових рядів.

Мета дипломної роботи - стислий опис деяких популярних моделей прогнозування часових рядів, що використовуються на практиці, з їх характерними рисами. Описані два можливі класи моделей часових рядів: стохастичні та нейронні мережі разом з їх притаманними для прогнозування сильними і слабкими сторонами. Відповідно метою є розробка програмного продукту для отримання практичних результатів та порівняння роботи результату прогнозування на основі нейронної мережі та стохастичних моделей. Також дано відповіді на питання пов'язаними з моделюванням часових рядів, такі як: стаціонарність, ошадливість, перенавчання і т.д.

Об'єктом дослідження є статистичні дані котирування акцій компаній, які використовуються для навчання нейронної мережі.

Abstract

The work consist of 76 pages 15 images 8 tables 19 sources

The theme: «Information system for prediction stock quotes on the basis of neural networks».

TIME SERIES, STOCHASTIC MODELS, SEASONALITY, ARTIFICIAL NEURAL NETWORKS.

Modeling and forecasting of time series is one of the most fundamental importance for various practical applications. In connection with this, in recent years there has been a lot of scientific work on this topic. The literature offers many important models for increasing the accuracy and efficiency of modeling and forecasting of time series.

The purpose of the thesis is a brief description of some popular models of forecasting of time series used in practice, with their characteristic features. Two possible classes of time series models are described: stochastic and neural networks, along with their inherent strengths and weaknesses for prediction. Accordingly, the goal is to develop a software product for obtaining practical results and to compare the work of the prediction result based on the neural network and stochastic models. Also, these answers to questions related to the simulation of time boards, such as: stationary, thrift, retraining, etc.

The subject of the study is the statistical data of the stock quotes of companies used to train the neural network.

ЗМІСТ

ВСТУП	9
РОЗДІЛ 1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ	10
1.1 Актуальність	10
1.2 Визначення часового ряду	12
1.3 Компоненти часового ряду	13
1.4 Введення в аналіз часових рядів	15
1.5 Часові ряди і стохастичний процес	16
1.6 Концепція стаціонарності	17
Висновки до розділу 1	19
РОЗДІЛ 2 МАТЕМАТИЧНІ ОСНОВИ РОБОТИ	21
2.1 Підходи до прогнозування часових рядів	21
2.2 Стохастичні моделі	21
2.2.1 Авторегресія з ковзним середнім	22
2.2.2 Аналіз стаціонарності	24
2.2.3 Функції автокореляції та часткової автокореляції	25
2.2.4 Авторегресія з інтегрованим ковзним середнім	26
2.2.5 Сезонна авторегресія з інтегрованим ковзним середнім	28
2.2.6 Деякі моделі нелінійних часових рядів	28
2.2.7 Методологія Бокса-Дженкінса	29
2.3 Прогнозування з використанням штучних нейронних мереж	31
2.3.1 Архітектура штучних нейронних мереж	32
2.3.2 Time Lagged Neural Networks (TLNN)	34
2.3.3 Сезонні штучні нейронні мережі	36

	7
2.4 Показники ефективності	37
Висновки до розділу 2	39
РОЗДІЛ 3 АРХІТЕКТУРА ТА АНАЛІЗ РЕЗУЛЬТАТІВ РОБОТИ	45
3.1 Короткий огляд	45
3.2 Щоквартальний набір даних по продажах	45
3.3 Щоквартальний набір даних з виробництва в США.....	48
Висновки до розділу 3	52
ВИСНОВКИ ПО РОБОТІ ТА ПЕРСПЕКТИВИ ПОДАЛЬШИХ ДОСЛІДЖЕНЬ	53

ВСТУП

Коли був створений ще перший фінансовий ринок, люди намагалися передбачати майбутні рухи цін, оскільки це обіцяло безліч переваг, таких як «прибуток з нізвідки» або захист капіталу від раптових рухів ринку. Актуальність ця проблема не втратила і до сих пір і, як відомо, є складною. Існує багато фінансових установ, інвестиційних фондів, банків і індивідуальних трейдерів, які намагаються передбачити ринок і знайти кращі моменти, для того щоб купити і продати, аби максимізувати прибуток.

В наш час проблема пов'язана з прогнозуванням котирування акцій здебільшого розв'язується за допомогою методів прогнозування на основі використання часового ряду однієї змінної, тобто на основі авторегресії, авторегресії з ковзним середнім (АРКС) та авторегресії з інтегрованим ковзним середнім та інші.

Питання полягає в наступному: чи можна поглянути на розв'язок даної задачі з кута нейронної мережі? Ми спостерігаємо за ринком і хочемо прийняти рішення: купувати, продавати або чекати. Якщо ми купуємо до того, як ціна зросте, наш прибуток буде позитивним, інакше ми отримаємо негативну винагороду. Нашим завданням є отримання якомога більшого прибутку.

Тоді ж які саме нейронні мережі найкращим чином підходять для даного завдання. Ґрунтуючись на отриманих результатах цієї нейромережі є нейромережа LSTM.

РОЗДІЛ 1

ОСНОВНІ ПОНЯТТЯ МОДЕЛЮВАННЯ ЧАСОВИХ РЯДІВ

1.1 Актуальність

Моделювання часових рядів - це динамічна дослідницька область, яка за останні десятиліття привернула увагу науковців. Основною метою дослідників даної області є ретельне збирання і ретельне вивчення минулих спостережень часових рядів для розробки відповідної моделі, яка описує структуру, властиву даному ряду. Ця модель потім використовується для генерації майбутніх даних для ряду, тобто для складання прогнозів. Таким чином, прогнозування часових рядів можна назвати актом прогнозування майбутнього шляхом розуміння минулого. Через неодмінну важливість прогнозування ряду часових рядів у багатьох практичних сферах, таких як бізнес, економіка, фінанси, наука та інженерія тощо очевидно, що успішне прогнозування часових рядів залежить від правильного налаштування моделі. Протягом багатьох років дослідниками була проведена велика робота по розробці ефективних моделей для підвищення точності прогнозування. В результаті в літературі з'явилися різні важливі моделі прогнозування часових рядів.

Однією з найбільш популярних і часто використовуваних моделей стохастичних часових рядів є модель авторегресійного інтегрованого ковзного середнього (АРІКС). Основне припущення, зроблене для реалізації цієї моделі, полягає в тому, що часовий ряд який даний за умови певного завдання є лінійним і дотримується певного відомого статистичного розподілу, наприклад - нормальний розподіл. Модель АРІКС має підкласи інших моделей, таких як авторегресивна (АР), ковзне середнє (КС) і авторегресивна з ковзним середнім (АРКС). Для прогнозування сезонних тимчасових рядів Джордж Бокс та Гвілім Дженкінс запропонували досить вдалу варіацію моделі АРІКС, а саме: сезонну АРІКС (САРІКС). Популярність моделі АРІКС

в основному обумовлена її гнучкістю і простотою подання декількох різновидів часових рядів, а також відповідною методологією Боксу-Дженкінса для оптимального процесу побудови моделі. Але серйозним обмеженням цих моделей є попередньо оцінена лінійна форма даного часового ряду, яка стає неадекватною в багатьох практичних ситуаціях. Для подолання цього недоліку в літературі були запропоновані різні нелінійні стохастичні моделі, однак з точки зору реалізації вони не такі прості і зрозумілі, як моделі АРІКС.

Останнім часом штучні нейронні мережі (ШНМ) привертають все більше уваги в області прогнозування часових рядів. Хоча спочатку вони були натхненні біологічними особливостями, але пізніше були успішно застосовані в багатьох різних областях, особливо в цілях прогнозування та класифікації. Відмінною особливістю ШНМ при застосуванні до задач прогнозування часових рядів є притаманна їм здатність нелінійного моделювання, без будь-яких припущень про статистичному розподілі, за яким слідують спостереження. Відповідна модель адаптивно формується на основі наведених даних. З цієї причини ШНМ за своєю природою засновані на даних і є самоадаптивними. За останні кілька років було виконано значний обсяг науково-дослідних робіт по застосуванню нейромереж для моделювання і прогнозування часових рядів. Сучасна дискусія про останні роботи по темі в нейронних мережах для прогнозування рядів зубів була представлена Чжан і ін. В 1998 г. У літературі представлені різні моделі прогнозування ШНМ. Найбільш поширеними і популярними серед них є багатошарові перцептрони (БП, multi-layer perceptrons MLP), які характеризуються одношаровою прихованої мережею прямої подачі (МПП, Feed Forward Network FNN). Інший широко використовуваним різновидом МПП є нейронна мережа з затримкою в часі (НМЗЧ, Time Lagged Neural Network TLNNN). У 2008 р К. Хамзачебі представив нову модель ШНМ, а саме модель сезонної штучної нейронної мережі (СШНМ, Seasonal Artificial Neural Network SANN) для прогнозування сезонних тимчасових рядів. Її запропонована модель дивно проста, а також пройшла експериментальну перевірку на предмет її успішності та

ефективності при прогнозуванні сезонних тимчасових рядів. Звичайно, в літературі є багато інших існуючих нейромережових структур, завдяки безперервній дослідницькій роботі в цій галузі.

1.2 Визначення часового ряду

Часові ряди являють собою послідовний набір точок даних, вимірюваних, як правило, протягом декількох послідовних періодів часу. Математично він визначається як набір векторів $x(t)$, $t = 0, 1, 2, \dots$ де t - час, що пройшов. Змінна $x(t)$ розглядається як випадкова змінна. Вимірювання, зроблені під час події в часовому ряді, розташовані в належному відсортовано-хронологічному порядку.

Тимчасові ряди, що містять записи однієї змінної, називаються одновимірними. Але, якщо розглядати записи більш ніж однієї змінної, то вона називається багатовимірної. Тимчасові ряди можуть бути неперервними або дискретними. У неперервних часових рядах спостереження вимірюються в кожному конкретному випадку часу, в той час як дискретні часові ряди містять спостереження, виміряні в дискретних точках часу. Наприклад, показники температури, течія річки, концентрація хімічного процесу і т.д. можуть бути записані як неперервні часові ряди. З іншого боку, населення конкретної країни місто, виробництво компанії, обмінні курси між двома різними валютами можуть являти собою приклади дискретного часового ряду. Зазвичай в дискретних тимчасових рядах записуються послідовні спостереження, через однакові проміжки часу, такі як годину, день, тиждень, місяць або рік. Як згадувалося раніше, змінна, яка спостерігається в дискретному часі ряду, являє собою наступне які передбачається вимірювати як неперервну змінну, використовуючи шкалу дійсних чисел. Крім того

неперервні часові ряди можуть бути легко перетворені в дискретні шляхом злиття даних разом протягом певного проміжку часу.

1.3 Компоненти часового ряду

Вважається, що на часовий ряд в цілому впливають чотири основних компоненти, які можуть бути відокремлені від спостережуваних даних. Цими компонентами є: тренд, сезонні, циклічні і випадковий шум. Короткий опис цих чотирьох компонент наводиться тут. Загальна тенденція до збільшення, зменшення або стагнації числового ряду протягом тривалого періоду часу називається трендом. Таким чином, можна сказати, що тренд це довгострокова характеристика часового ряду. Наприклад, ряд показників, що стосуються зростання чисельності населення, кількості осіб, що мають відношення до приросту чисельності населення. будинки в місті і т.д. мають тенденцію до зростання, в той час як низхідна тенденція простежується послідовно. пов'язані з рівнем смертності, епідеміями і т.д.

Сезонні коливання часового ряду - це коливання протягом року, протягом сезону. Важливими чинниками, що викликають сезонні коливання, є: клімат і погодні умови, звичаї, традиційні звички і т.д. Наприклад, влітку збільшується продаж морозива, взимку - продажу вовняних тканин. Сезонні коливання є важливим фактором для підприємців, власників магазинів і виробників при складанні правильних планів на майбутнє. Циклічна варіація в часі ряду описує середньострокові зміни в цьому ряду, викликані обставинами, які повторюються в циклах. Довжина циклу триває більш протяжний період часу, зазвичай два або більше років. Більшість економічних і фінансових часових рядів мають циклічні коливання. Наприклад, діловий цикл складається з чотирьох етапів, а саме:

i) процвітання, ii) спад, iii) депресія і iv) відновлення.

Схематично типовий бізнес цикл представлений рисунком 1.1



Рисунок 1.1-Типовий бізнес цикл

Ці коливання викликані наступними причинами такі, як війна, страйк, землетрус, повінь, революція і т.д. Певного статистичного методу для вимірювання випадкових флуктуацій в часовому ряді не існує.

З огляду на вплив цих чотирьох компонентів, для часового ряду зазвичай використовуються дві різні типи моделей, а саме мультиплікативна і адитивна моделі.

Мультиплікативна модель:

$$Y(t) = T(t) \times S(t) \times S(t) \times C(t) \times I(t)$$

Адитивна модель:

$$Y(t) = T(t) + S(t) + S(t) + C(t) + I(t)$$

де $Y(t)$ – спостереження;

$T(t)$ – тренд;

$S(t)$ – сезонність;

$C(t)$ – циклічність;

$I(t)$ - випадковий шум в часі t .

Мультиплікативна модель заснована на припущенні, що чотири компоненти часового ряду не обов'язково незалежні і можуть впливати один на одного, тоді як в адитивній моделі передбачається, що ці чотири компоненти незалежні одна від одної.

1.4 Введення в аналіз часових рядів

На практиці певна модель відповідає певному часовому ряду і розраховується з використанням відомих значень даних. Процедура підгонки часового ряду під певну правильну модель називається аналізом часового ряду. Аналіз включає в себе методи, які намагаються зрозуміти природу тренду і є корисними для майбутнього прогнозування і моделювання.

При прогнозуванні часових рядів минулі спостереження збираються і аналізуються для розробки відповідної математичної моделі, яка фіксує основний процес генерації даних для тренду. Потім за допомогою цієї моделі прогнозуються майбутні події. Такий підхід особливо корисний в тих випадках, коли мало що відомо про статистичний шаблон, за яким слідують послідовні спостереження, або коли відсутня задовільна підходяща модель. Прогнозування часових рядів має важливе застосування в різних областях. Часто за результатами прогнозу приймаються цінні стратегічні рішення і коригувальні заходи. Таким чином, складання хорошого прогнозу, тобто відповідність адекватної моделі тимчасовому ряду може бути різним. За останні кілька десятиліть дослідниками було зроблено багато зусиль для розробки та вдосконалення відповідних моделей прогнозування часових рядів.

1.5 Часові ряди і стохастичний процес

Часові ряди є недетермінованими за своєю природою, тобто ми не можемо з упевненістю передбачити, що станеться в майбутньому. Зазвичай вважається, що часовий ряд $\{x(t), t = 0, 1, 2, \dots\}$ слідує певній ймовірнісній моделі, яка описує розподіл випадкової змінної x_t . Математичне визначення, що описує вірогідну структуру часового ряду, називається стохастичним процесом. Таким чином, послідовність спостережень ряду фактично є прикладом реалізації стохастичного процесу, який породив його.

В основному передбачається, що змінні часового ряду x_t є незалежними і однаково розподіленими відповідно до нормального розподілу. Однак, як уже згадувалося, цікавим моментом є те, що часові ряди насправді є не зовсім незалежними і однаково розподіленими; вони сліднують більш-менш визначеного шаблону в довгостроковій перспективі. Наприклад, якщо сьогодні температура повітря в певному місті є надзвичайно висока, то можна обґрунтовано припустити, що завтрашня температура також може бути високою.

Стохастичним процесом називається процес, який розвивається в часі відповідно до законів теорії ймовірностей. До стохастичних процесів належить клас стаціонарних процесів. Стохастичний процес називається стаціонарним, якщо його основні властивості залишаються незмінними в часі. Припустимо, що досліджується часовий ряд x . Позначимо через $x(t)$, рівень даного часового ряду. Тоді стаціонарний процес буде характеризуватися такими чотирма властивостями:

- 1) математичне сподівання стаціонарного ряду $E(y_t)$, є постійним, тобто середнє значення часового ряду, навколо якого змінюються рівні, є величиною постійною:

$$E(y_t) = \bar{y} = \text{const};$$

- 2) дисперсія стаціонарного ряду є постійною. Вона характеризує варіацію рівнів часового ряду щодо його середнього значення

$$\bar{x}: D(y) = E(y_t - \bar{y})^2 = G^2(y) = \text{const}$$

- 3) автоковаріація стаціонарного ряду з лагом l є сталою, тобто коваріація між значеннями x_t і x_{t+l} , відокремленими інтервалом в l одиниць часу, визначається за формулою:

$$R_l(y_t) = \text{cov}(y_t, y_{t+l}) = E[(y_t - \bar{y})(y_{t+l} - \bar{y})];$$

для стаціонарних рядів автоковаріація залежить тільки від величини лага l , тому справедливо рівність виду:

$$R_{l=0}(y_t) = G^2(y);$$

- 4) коефіцієнти автокореляції стаціонарного ряду з лагом l є постійними. Отже, автокореляція є нормованої автоковаріацією, оскільки Для стаціонарного процесу $G^2(y) = \text{const}$:

$$\rho_l = \frac{E[(y_t - \bar{y})(y_{t+1} - \bar{y})]}{\sqrt{E(y_t - \bar{y})^2 E(y_{t+1} - \bar{y})^2}} = \frac{E[(y_t - \bar{y})(y_{t+1} - \bar{y})]}{G^2(y)}$$

Таким чином, коефіцієнт автокореляції порядку l визначається за формулою:

$$\rho_l = \frac{R_l(y_t)}{R_{l=0}(y_t)}$$

Нестационарним часовим рядом називається ряд, який не задовольняє перерахованим вище властивостям.

Випадковий процес, званий білим шумом, є окремим випадком стаціонарних часових рядів. Білим шумом називається випадкова послідовність значень y_1, y_2, \dots, y_N , якщо її математичне сподівання дорівнює нулю, тобто

$$E(y_t) = 0$$

$$\text{де } t = \overline{1, N}.$$

Елементи є некорельованими (незалежними один від одного) однаково розподіленими величинами, і дисперсія є постійною величиною яка визначається

$$D(y_t) = G^2(y) = \text{const.}$$

Білий шум - це теоретичний процес, який реально не існує, проте він представляє собою дуже важливу математичну модель, яка використовується при розв'язанні безлічі практичних завдань.

1.6 Концепція стаціонарності

Концепція стаціонарності стохастичного процесу можна візуалізувати як форму статистичної рівноваги. Статистичні властивості, такі як математичне сподівання і дисперсія стаціонарного процесу, які не залежать від часу. Це необхідна умова для побудови моделі часових рядів, що

використовується в подальшому для прогнозування. Крім того, математична складність підігнаної моделі зменшується з цим припущенням. Існує два типи стаціонарних процесів, які визначені нижче:

- Процес $\{x(t), t = 0, 1, 2, \dots\}$ є сильно стаціонарним або строго стаціонарним, якщо спільна функція розподілу ймовірностей $\{x_{t-s}, x_{t-s+1}, \dots, x_t, \dots, x_{t+s-1}, x_{t+s}\}$ не залежить від t для всіх s . Таким чином, для строгого стаціонарного процесу спільний розподіл довільної кількості можливих випадкових величин змінні процесу не залежать від часу.
- Однак для практичного застосування допущення про строгую стаціонарність не завжди необхідно і тому розглядається слабша форма. Вважається, що стохастичний процес є слабким стаціонарним k -го порядку, якщо статистичні моменти процесу до цього порядку залежать тільки від різниці в часі, а не від часу появи даних, що використовуються для оцінки моментів. Наприклад, стохастичний процес $\{x(t), t = 0, 1, 2, \dots\}$ є стаціонарним процесом другого порядку, якщо він має математичне сподівання і дисперсію, незалежну від часу, і коваріаційні значення $Cov(x_t, x_{t-s})$ залежать тільки від s .

Важливо відзначити, що ні сильна, ні слабка стаціонарність не мають на увазі один одного. Однак, слабо стаціонарний процес після нормального розподілу також є сильно стаціонарним. Деякі математичні тести, подібні до того, що дали Діккі і Фуллер, зазвичай використовуються для виявлення стаціонарності в даних часових рядів.

Як згадувалося в роботах [6, 23], поняття стаціонарності - це математична ідея, побудована для спрощення теоретичного і практичного розвитку стохастичних процесів. Щоб створити правильну та адекватну для майбутнього прогнозування, базовий тимчасовий ряд, як очікується, буде стаціонарним. На жаль, це не завжди так. Як стверджують Хайпель і Маклеод [23], чим більша ймовірність того, що часові ряди будуть відображатися протягом певного періоду часу, тим більша ймовірність того, що будуть

присутні нестационарні характеристики. Однак протягом відносно короткого проміжку часу можна розумно змодельовати тренд, використовуючи стаціонарний стохастичний процес. Зазвичай це часові ряди, що показують тенденцію або сезонний характер носить нестационарний характер. У таких випадках, відмінності в поглядах і повноваженнях перетворення часто використовуються для видалення тренда і додання серії стаціонарності.

Висновки до розділу 1

При побудові правильної моделі часових рядів необхідно враховувати принцип Парсімона [22]. Відповідно до цього принципу завжди вибирається модель з найменшою кількістю параметрів, щоб забезпечити адекватне уявлення вихідних даних часових рядів. З безлічі відповідних моделей слід розглядати найпростішу, зберігаючи при цьому точний опис властивостей, властивих часовому ряду. Ідея модельної стриманості схожа на відомий бритвенний принцип Оккама [23]. Як обговорювали Hipel і McLeod [23], один з аспектів цього принципу полягає в тому, що, зіткнувшись з рядом конкуруючих і адекватних пояснень, вибирайте найпрості. Бритва Оккама надає значну внутрішню інформацію при застосуванні до логічного аналізу.

Більш того, чим складніша модель, тим більше можливостей для відхилення від фактичних припущень моделі. Зі збільшенням параметрів моделі, ризик переобладнання також зростає. Відповідна модель часових рядів може дуже добре описувати дані навчання, але вона може виявитися непридатною для майбутнього прогнозування. Оскільки потенційне переоснащення впливає на здатність моделі добре прогнозувати, економія часто використовується в якості орієнтира задля вирішення цієї проблеми.

Таким чином, підбиваючи підсумок, можна сказати, що, слід приділяти увагу вибору найбільш скупой моделі серед всіх інших варіантів.

РОЗДІЛ 2

МАТЕМАТИЧНІ ОСНОВИ РОБОТИ

2.1 Прогнозування часових рядів з використанням різних підходів

Більшість існуючих методів побудови прогнозування часових рядів можна умовно розділити на дві категорії:

- на основі стохастичних моделей.
- на основі штучних нейронних мереж (ШНМ).

2.2 Стохастичні моделі

У попередньому розділі описані основи моделювання часових рядів і прогнозування. Вибір правильної моделі надзвичайно важливий, оскільки вона відображає базову структуру рядів і ця підходяща модель, в свою чергу, використовується для прогнозування майбутнього. А модель часового ряду називається лінійною або нелінійною залежно від того, чи поточне значення ряду є лінійною або нелінійною функцією минулих спостережень.

В цілому моделі для даних часових рядів можуть мати безліч форм і представляти собою різні моделі стохастичні процеси. У літературі широко використовуються дві моделі лінійних часових рядів, а саме: Моделі з авторегресії (АР) і Ковзного середнього (КС). Об'єднуючи ці дві моделі отримуємо - авторегресія з ковзним середнім (АРКС) і авторегресія з інтегрованим ковзним середнім (АРІКС), які були запропоновані в літературі [12,23]. Модель авторегресійна фракційно-інтегрована ковзного середнього (ARFIMA) узагальнює моделі АРКС і АРІКС. Для прогнозування сезонних тимчасових рядів використовується варіація моделі АРІКС, тобто Сезонна авторегресивна з інтегрованим ковзним середнім (САРІКС). Модель АРІКС і

її різні варіанти засновані на знаменитому принципі Бокса-Дженкінса, тому вони також широко відомі як моделі Бокса-Дженкінса. Лінійні моделі привернули велику увагу завдяки своїй відносній простоті в розумінні і реалізації. Однак багато практичних часових рядів показують нелінійні закономірності. Наприклад, як зазначив Р. Парреллі в [28], нелінійні моделі підходять для прогнозування змін волатильності в економічних і фінансових часових рядах. З огляду на ці факти, в літературі пропонуються різні нелінійні моделі. Деякі з них - знамениті авторегресивні. Модель умовної гетероскедастичності (ARCH) [9, 28] і її різновиди як Узагальнена ARCH (GARCH) [9, 28], Експоненційно узагальнена ARCH (EGARCH) [9] та ін. Модель нелінійної авторегресії (NAR) [7], модель нелінійного ковзного середнього (NMA) [28] та ін.

2.2.1 Авторегресія з ковзним середнім

Модель Авторегресії з ковзним середнім – АРКС (ARMA) (p, q) являє собою комбінацію моделей АР(AR) (p) і КС(MA) (q) і підходить для моделювання одновимірних часових рядів. У моделі АР (p) передбачається, що нове значення змінної буде лінійною комбінацією p минулих спостережень і випадкової похибки в поєднанні з постійною термін. Математично модель АР (p) може бути виражена як :

$$y_t = c + \sum_{i=1}^p \varphi_i y_{t-i} + \varepsilon_t = c + \varphi_1 y_{t-1} + \varphi_2 y_{t-2} + \dots + \varphi_p y_{t-p} + \varepsilon_t$$

Тут y_t і ε_t є відповідно фактичними значеннями і випадковою помилкою (або випадковим ударом) на момент часу. період t , φ_i ($i = 1, 2, \dots, p$) -

параметри моделі і c - константа. Цілочисельна константа p називається порядком проходження моделі. Іноді константа опускається для простоти.

Подібно до того, як AR (p) модель регресує по відношенню до минулих значень серії, MA (q) модель використовує минулі значення. помилки як пояснюють змінних. Модель MA (q):

$$y_t = \mu + \sum_{j=1}^q \theta_j \varepsilon_{t-j} + \varepsilon_t = \mu + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \dots + \theta_p \varepsilon_{t-q} + \varepsilon_t$$

де μ - середнє значення ряду;

$\theta_j (j = 1, 2, \dots, q)$ = параметри моделі;

q - параметри моделі порядок проходження моделі.

Випадковими ударами вважаються білі шуми, тобто послідовність незалежних і однаково розподілених випадкових (тобто d) змінних з нульовим середнім значенням і постійною дисперсією. Зазвичай передбачається, що випадкові поштовхи слідує за типовими шоками нормального розподілу. Таким чином, концептуально модель ковзного середнього являє собою лінійну регресію поточне спостереження часового ряду проти випадкових поштовхів одного або більше попередніх часових рядів. Підгонка MA-моделі до часових рядів складніше, ніж підгонка AR, тому що в першому випадку випадкові члени помилки не передбачуються.

Моделі з авторегресії (AR) і ковзним середнім (MA) можуть бути ефективно об'єднані разом. для формування загального і корисного класу моделей часових рядів, відомих як моделі ARMA.

Математично модель ARMA (p, q) представлена як:

$$y_t = c + \varepsilon_t + \sum_{i=1}^p \varphi_i y_{t-i} + \sum_{j=1}^q \theta_j \varepsilon_{t-j}$$

Тут модель впорядковує p, q відповідно до p авторегресії і q умовами ковзного середнього.

Зазвичай моделі ARMA змінюються за допомогою нотації лагового оператора. Оператор запізнювання або зворотного зсуву визначається як $Ly_t = y_{t-1}$. Поліноми лагового оператора або лагові поліноми використовуються для представлення моделей ARMA наступним чином:

AR(p) модель:

$$\varepsilon_t = \varphi(L)y_t$$

MA(q) модель:

$$y_t = \theta(L)\varepsilon_t$$

ARMA(p, q) модель:

$$\varphi(L)y_t = \theta(L)\varepsilon_t$$

$$\begin{aligned} \text{де } \varphi(L)y_t &= \sum_{i=1}^p \varphi_i L^i; \\ \theta(L) &= 1 + \sum_{j=1}^q \theta_j L_j. \end{aligned}$$

Показано, що важливою властивістю процесу AR (p) є оборотність, тобто процес AR (p) завжди може бути записаний у вигляді процесу MA (∞). Якщо MA (q) процес повинен бути інвертованим, то всі корені рівняння $\theta(L) = 0$ повинні лежати поза одиничним кола. Ця умова відома як умова інвертованості для процесу MA.

2.2.2 Аналіз стаціонарності

Якщо процес AR (p) представлений у вигляді $\varepsilon_t = \varphi(L)y_t$, то $\varphi(L) = 0$ відомий під назвою характеристичне рівняння для процесу. Box і Jenkins [23] доводять, що необхідна і достатня умова для того, щоб процес AR (p) був стаціонарним, є те, що всі корені процесу характеристичного рівняння має виходити за межі одиничного кола. Хайпель і Маклеод [23] згадали про існування іншого простого алгоритму (Шур і Пагано) для визначення стаціонарності процесу AR.

Наприклад, як показано в [12] AR (1) модель

$$y_t = c + \varphi_1 y_{t-1} + \varepsilon_t$$

є стаціонарною, коли $|\varphi_1| < 1$ з

Процес MA (q) завжди є стаціонарним, незалежно від значень параметрів MA [23]. Умови щодо стаціонарності і інвертованості процесів AR і MA також зберігаються для процесу ARMA. Процес ARMA (p, q) є стаціонарним, якщо всі корені характеристичного рівняння $\varphi(L) = 0$ лежать поза одиничного кола. Аналогічно, якщо всі корені лагового рівняння $\theta(L) = 0$ лежать за межами одиничного кола, то процес ARMA (p, q) є інвертованим і може бути виражений у вигляді чистого AR-процесу.

2.2.3 Функції автокореляції та часткової автокореляції

Для визначення правильної моделі для даних даного часового ряду необхідно провести аналіз автокореляційної функції (АКФ) і частково

актокореляційної функції (ЧАКФ). Ці статистичні показники відображають те, як спостереження за певний період часу пов'язані один з одним. Для цілей моделювання та прогнозування часто буває корисно побудувати графік АКФ і ПАКФ з урахуванням послідовних часових лагів. Ці графіки допомагають визначити порядок проходження термінів AR і MA. Нижче наводяться їх математичні визначення:

Для часових рядів $\{x(t), t = 0, 1, 2, \dots\}$ Автоковаріацію з лагом k визначають наступним чином:

$$\gamma_k = Cov(x_t, x_{t+k}) = E[(x_t - \mu)(x_{t+k} - \mu)]$$

Автокореляційний коефіцієнт з лагом k визначають наступним чином:

$$\rho_k = \frac{\gamma_k}{\gamma_0}$$

де μ - середнє значення часового ряду, тобто $\mu = E[x_t]$

Автоковаріація при лагу 0, тобто γ_0 є різницею в часі ряду. З визначення зрозуміло, що коефіцієнт автокореляції ρ_k не має розмірів і тому не залежить від масштабу виміру. Також очевидно, що $1 \leq \rho_k \leq 1$.

Статистики Вох і Дженкінс назвали γ_k теоретичною автоковаріаційною функцією і ρ_k як теоретично автокореляційною функцією (АКФ).

Інша міра, відома як функція часткової автокореляції (ЧАКФ), використовується для вимірювання кореляції між k періодами спостережень тому і поточним наглядом, після контролю для спостережень з проміжними лагом (тобто з лагом $< k$). При лагу рівному 1, ЧАКФ (1) такий як і АКФ (1). Більш розширені формули розрахунку ЧАКФ наведені в [6, 23].

Зазвичай стохастичний процес, керуючий часовими рядами, невідомий, і, отже, неможливо визначити фактичні або теоретичні значення АКФ і ЧАКФ.

Швидше, вони повинні бути розрахованими на основі даних про навчання, тобто на основі відомих часових рядів.

Як зазначено в [23], найбільш підходящою вибірковою оцінкою для Автоковаріаційної Функції з лагом k є

$$c_k = \frac{1}{n} \sum_{t=1}^{n-k} (x_t - \mu)(x_{t+k} - \mu)$$

Потім оцінка для вибірки АКФ з лагом k виглядає так:

$$r_k = \frac{c_k}{c_0}$$

де $\{x(t), t = 0, 1, 2, \dots\}$ тренувальний часовий ряд n із середнім μ ;

Як пояснили Box і Jenkins [6], зразок графіка АКФ корисний для визначення типу часового ряду довжиною N . Оскільки АКФ симетричний приблизно нульового лагу, то це тільки для побудови вибірки АКФ для позитивних лагів, починаючи з лага одного і закінчуючи максимальним лагом (приблизно $N/4$).

2.2.4 Авторегресія з інтегрованим ковзним середнім

Моделі ARMA, описані вище, можуть використовуватися тільки для даних стаціонарних часових рядів. Однак на практиці багато часових рядів, наприклад, відносяться до соціально-економічних і бізнесових та проявляє нестаціонарну поведінку. Часові ряди, які містять тренди і сезонні моделі, є

також нестационарними за своїм характером. Таким чином, з точки зору застосування моделі ARMA є недостатньою для належного опису нестационарних часових рядів, які часто зустрічаються. З цієї причини пропонується модель ARIMA, яка є узагальненим варіантом моделі ARMA, щоб включити і випадок нестационарності. У моделях ARIMA нестационарні часові ряди стають стаціонарними шляхом нанесення кінцевого значення відмінності в точках даних. Математичне формулювання моделі ARIMA (p, d, q) з використанням лагових поліномів наведено нижче:

$$\varphi(L)(1-L)^d y_t = \theta(L)\varepsilon_t$$

Тобто,

$$\left(1 - \sum_{i=1}^p \varphi_i L^i\right) (1-L)^d y_t = \left(1 + \sum_{j=1}^q \theta_{j=1} L^j\right) \varepsilon_t$$

- Тут p, d і q є цілими числами, які є більшими або рівними нулю і відносяться до порядку авторегресії, інтегрованості і ковзного середнього даної моделі відповідно.

- Ціле число d контролює рівень відмінностей. Зазвичай в більшості випадків досить d=1. Коли d=0, то вона зменшується до моделі ARMA (p, q).

- ARIMA (p, 0,0) - це ніщо інше, як AR (p) модель, а ARIMA (0,0, q) - MA (q) модель.

- ARIMA (0,1,0), тобто $y_t = y_{t-1} + \varepsilon_t$ є спеціальною і відома як модель Random Walk. [8, 12, 21]. Вона широко використовується для нестационарних даних, таких як економічні ряди і часові ряди для котирування акцій.

Корисним узагальненням моделей ARIMA є Авторегресійна з дрібно-інтегрованим ковзним середнім (ARFIMA), яка допускає не числові значення

різниці d . ARFIMA має корисне застосування в моделюванні часових рядів з великою пам'яттю [17].

У даній моделі розширення терміну $(1 - L)^d$ має бути зроблено з використанням загальної біноміальної теореми.

2.2.5 Сезонна авторегресія з інтегрованим ковзним середнім

Модель ARIMA призначена для несезонних нестационарних даних. Бокс і Дженкінс узагальнили дану модель для обліку сезонності. Пропонована ними модель відома як модель Сезонна ARIMA (SARIMA). У цій моделі сезонна диференціація відповідного порядку використовується для видалення нестационарності з часового ряду. Сезонна різниця першого порядку - це різниця між спостереженням даного часу і відповідним спостереженням в попередньому році і розраховується як $z_t = y_t - y_{t-s}$. Для місячних часових рядів $s = 12$ та для квартальних часових рядів $s = 4$. Ця модель зазвичай називається моделлю s SARIMA $(p, d, q) \times (P, D, Q)^s$.

Математичне формулювання моделі SARIMA в термінах лагових поліномів наведено нижче.

$$\Phi_p(L^s)\varphi_p(L)(1 - L)^d(1 - L^s)^D y_t = \Theta_Q(L^s)\theta_q(L)\varepsilon_t$$

Тобто

$$\Phi_p(L^s)\varphi_p(L)z_t = \Theta_Q(L^s)\theta_q(L)\varepsilon_t$$

де z_t - ряди з сезонною відмінністю

2.2.6 Деякі моделі нелінійних часових рядів

До сих пір було описано моделі лінійних часових рядів. Як згадувалося раніше, нелінійні моделі також слід розглядати для кращого аналізу та прогнозування часових рядів. Кемпбелл, Ло і Маккінлі (1997 г.) внесли важливий внесок в цей напрям. На їхню думку, практично всі нелінійні тимчасові ряди можна розділити на дві гілки: одна включає в себе моделі середнього нелінійного, а інша включає моделі нелінійної дисперсії (гетероскедастичних).

2.2.7 Методологія Бокса-Дженкінса

Після опису різних моделей часових рядів наступним питанням, яке турбує, є те, як вибрати належну модель, яка може дати точний прогноз, заснований на описі історичних подій, шаблон даних і як визначити оптимальні моделі. Статистики Джордж Бокс і Гвілім Дженкінс [6] розробили практичний підхід до побудови моделі ARIMA, яка найкращим чином підходить для вирішення цього завдання до певного часового ряду, а також задовольняють принципом скупості. Їх концепція має фундаментальне значення в області аналізу і прогнозування часових рядів.

Методологія Боксу-Дженкінса не передбачає будь-якої конкретної закономірності в історичних даних ряду, які необхідно спрогнозувати. Швидше можна сказати, що вона використовує триступеневий ітеративний підхід ідентифікації моделі, оцінки параметрів і діагностичної перевірки для визначення кращої економною моделі із загального класу ARIMA моделей.

Цей триступеневий процес повторюється кілька разів, поки остаточно не буде обрана задовільна модель. Потім ця модель може бути використана для прогнозування майбутніх значень часового ряду.

Метод прогнозування Боксу-Дженкінса схематично показаний на рисунку 2.1



Рисунок 2.1-Метод прогнозування Бокса-Дженкінса

Найважливішим кроком при виборі відповідної моделі є визначення оптимальних параметрів даної моделі. Одним з критеріїв є те, що вибірка АКФ і ПАКФ, розрахована на основі даних навчання повинні збігатися з відповідними теоретичними або фактичними значеннями. Інші широко відомі критерії для ідентифікації моделі є Акайке (AIC):

$$AIC(p) = n * \ln(\widehat{\sigma}_e^2 / n) + 2p$$

і Байєсівський інформаційний критерій (БІК):

$$BIC(p) = n * \ln(\widehat{\sigma}_e^2 / n) + p + p \ln(n)$$

де n - кількість ефективних спостережень, які використовуються для підгонки моделі;

p - число в моделі;

$\widehat{\sigma}_e^2$ - це сума квадратних залишків зразка.

Оптимальний порядок розташування моделей вибирається за кількістю параметрів моделі, що мінімізує або AIC, або BIC. Інші аналогічні критерії також були запропоновані в літературі для оптимальної ідентифікації моделі..

2.3 Прогнозування з використанням штучних нейронних мереж

У попередньому розділі ми обговорили важливі стохастичні методи моделювання та прогнозування часових рядів. Штучні нейронні мережі (ШНМ) були запропоновані в якості альтернативи прогнозування часових рядів і придбали величезну популярність в останні кілька років. Основне завдання ШНМ полягала в побудові моделі імітації інтелекту людського мозку в машину. Подібно роботі людського мозку, ШНМ намагаються розпізнати закономірності у вхідних даних, винести уроки з досвіду і потім надати узагальнені результати, засновані на їх отриманих раніше знаннях.

Хоча розробка ШНМ була в основному мотивована біологічними причинами, але згодом вони застосовуються в багатьох різних областях, особливо для прогнозування і класифікації мети. Найбільш характерними особливостями ШНМ, які роблять їх вельми улюбленими для аналізу і прогнозування часових рядів наведено нижче.

По-перше, ШНМ за своєю природою засновані на даних і є самоадаптивними. Немає необхідності вказувати конкретну форму моделі або робити апріорні припущення про статистичному розподілі даних, бажана модель адаптивно формується на основі особливостей, представлених на основі даних. Такий підхід дуже корисний в багатьох практичних ситуаціях, коли теоретичних рекомендацій немає необхідних для відповідного процесу генерації даних.

По-друге, ШНМ за своєю природою є нелінійними, що робить їх більш практичними і точними в наступних областях моделювання складних моделей даних на відміну від різних традиційних лінійних підходів, таких як методи ARIMA. Існує безліч прикладів, які свідчать про те, що ШНМ зробили набагато кращий аналіз і прогнозування, ніж різні лінійні моделі.

Нарешті, на думку Хорніка і Стінчкомб [22], ШНМ є універсальними функціональними апроксиматорами. Вони показали, що мережа може наблизити будь-яку неперервну функцію до будь-якої необхідної точності [5, 22]. ШНМ використовують паралельну обробку інформації з даних для апроксимації великого класу функцій з високим ступенем точності. Крім того, вони можуть мати справу з ситуацією, коли вхідні дані помилкові, неповні або розпливчасті [20].

2.3.1 Архітектура штучних нейронних мереж

Найбільш широко використовуваними ШНМ для прогнозування проблем є багатошарові перцептрони (БП), що використовують одну приховану шарну мережу прямої подачі (ШСП) [5,8]. Модель характеризується мережею з трьох рівнів, тобто вхідного, прихованого і вихідного, з'єднаних ациклічними зв'язками. Може бути більш одного прихованого шару. Вузли в різних шарах також відомі як обробні елементи. Трирівнева архітектура ШНМ для прямої передачі даних може бути представлена у вигляді діаграми, як показано на рисунку 2.2:

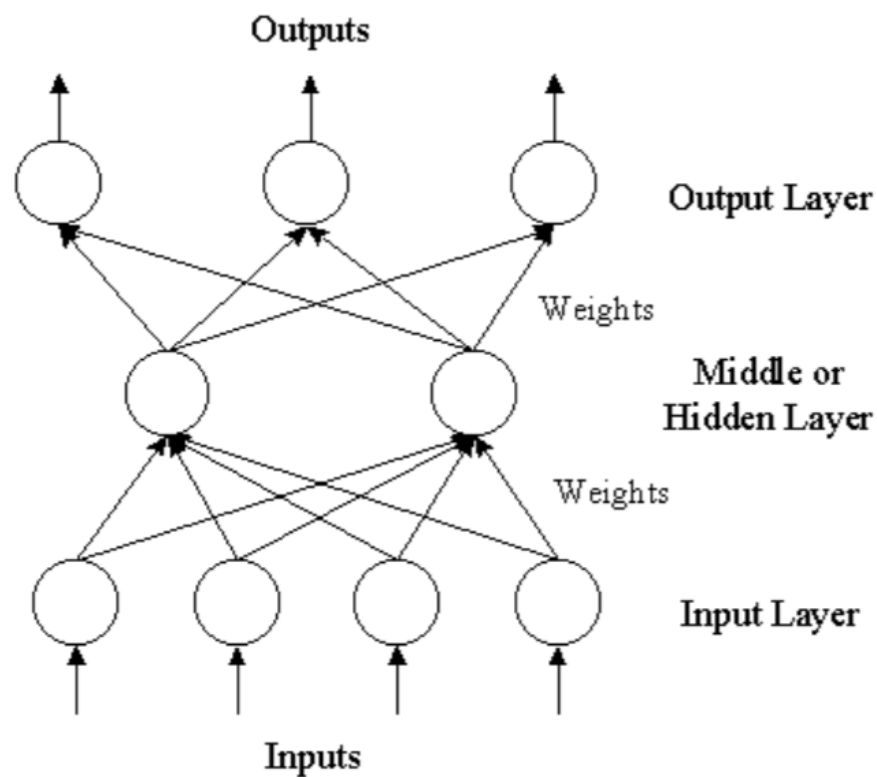


Рисунок 2.2-Трирівнева архітектура ШНМ для прямої передачі даних

Результат моделі обчислюється з використанням наступного математичного виразу

$$y_t = a_0 + \sum_{j=1}^q a_j g \left(\beta_{0j} + \sum_{i=1}^p \beta_{ij} y_{t-i} \right) + \varepsilon_t, \forall t$$

де $y_{t-i} (i = 1, 2, \dots, p)$ є p входами і y_t – виходами;
 p, q – це цілі числа, яке рівні числу вхідних і прихованих вузлів відповідно;
 $\alpha_j (j = 0, 1, 2, \dots, q)$ і $\beta_{ij} (i = 0, 1, 2, \dots, p; j = 0, 1, 2, \dots, q)$ ваги зв'язку;
 ε_t – білий шум;
 α_0 і β_{0j} - члени зміщення.

Зазвичай логістична сігмоїдна функція $g(x) = \frac{1}{1+e^{-x}}$ використовується в якості нелінійної функції активації. Також можуть використовуватися інші функції активації, такі як лінійна, гіперболічна, гаусівська та ін.

Пряма ШНМ модель фактично виконує нелінійне функціональне відображення від попередніх спостережень часового ряду до майбутнього значенням, тобто

$$y_t = f(y_{t-1}, y_{t-2}, \dots, y_{t-p}, w) + \varepsilon_t$$

де w - вектор всіх параметрів;
 f - функція, яка визначається структурою мережі і вагами з'єднань.

Для оцінки вагових коефіцієнтів підключення використовуються нелінійні найменш квадратні процедури, а саме виходячи з мінімізації функції помилки

$$F(\Psi) = \sum_t e_t^2 = \sum_t (y_t - \hat{y}_t)^2$$

де Ψ простір всіх вагових коефіцієнтів підключення.

Методи оптимізації, використовувані для мінімізації функції помилки називаються Правила навчання. Найвідомішим правилом навчання в літературі є правило зворотнього поширення або узагальнене дельта правило

2.3.2 Time Lagged Neural Networks (TLNN)

У вищеописаній формулюванні FNN вхідними вузлами є послідовні спостереження часового ряду, тобто x_t – функцією зі значеннями x_{t-i} , ($i = 1, 2, \dots, p$) де p - кількість вхідних вузлів. Також широко використовується інший варіант FNN, а саме архітектура TLNN. У TLNN вхідними вузлами є значення часових рядів з певними запізнюваннями. Наприклад, типовий TLNN для тимчасового ряду з сезонним періодом $s = 12$ може містити вхідні вузли в ролі яких будуть виступати лагові значення в час $t - 1, t - 2$ і $t - 12$. Значення в момент часу t має бути спрогнозовано з використанням значень з запізненням 1, 2 і 12. Представлена архітектура для щомісячних даних на рисунку 2.3

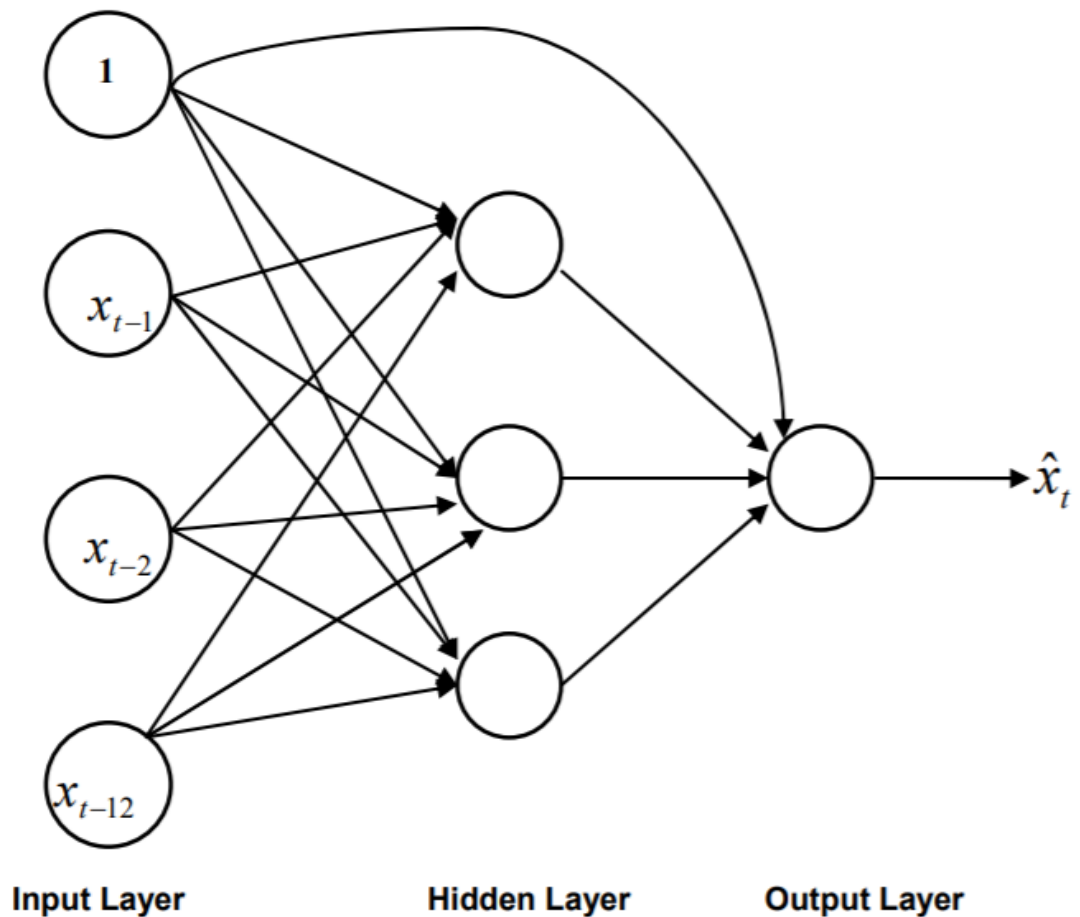


Рисунок 2.3-Типова архітектура TLNNN для щомісячних даних

Крім того, є постійний термін введення, який зручно сприймається як 1, а саме підключених до кожного нейрона в прихованому і вихідному шарах. Введення цієї постійної одиниці дозволяє уникнути необхідності окремо вводити термін зміщення.

Для TLNNN з одним прихованим рівнем загальне рівняння прогнозування для розрахунку прогнозу можна записати так:

$$\hat{x}_t = \phi_0 + \left\{ w_{c0} + \sum_h w_{h0} \phi_h \left(w_{ch} + \sum_i w_{ih} x_{t-j_i} \right) \right\}$$

де $x_{t-j_1}, x_{t-j_2}, \dots, x_{t-j_k}$ вхідні значення;

$\{w_{ch}\}$ є вагами для з'єднань між постійним входом і прихованими нейронами;

ϕ_h і ϕ_0 є відповідно рухами активації вихідного і вхідного нейронів.

Також варто пам'ятати, що w_{c0} є вагою прямого зв'язку між постійним входом і виходом. Також, $\{w_{ih}\}$ і $\{w_{h0}\}$ позначають ваги для інших зв'язків між вхідними та вихідними нейронами і між одне одним відповідно.

2.3.3 Сезонні штучні нейронні мережі (SANN)

Структура SANN запропонована К. Хамзачебі для поліпшення продуктивності прогнозування ШНМ для сезонних часових рядів. Пропонована модель SANN не передбачає вимаги попередньої обробки необроблених даних. Також SANN може ознайомитися з сезонним малюнком в цьому ряду, не усуваючи їх, на відміну від деяких інших традиційних підходів, таких як SARIMA, яка була проілюстрована в розділі 2. Автор емпірично підтвердив хорошу здатність прогнозувати SANN на чотирьох практичних наборах часових рядів.

У цій моделі сезонний параметр s використовується для визначення кількості споживаних ресурсів і нейронів на виході. Ця обставина робить модель дивно простою для розуміння і реалізації. Як значення вхідних і вихідних нейронів в цій мережевій структурі використовуються спостереження сезонного періоду i^{th} і $(i + 1)^{th}$. Кожен сезонний період складається з ряду спостережень. Схематично структура SANN може бути представлена у вигляді рисунку 2.4:

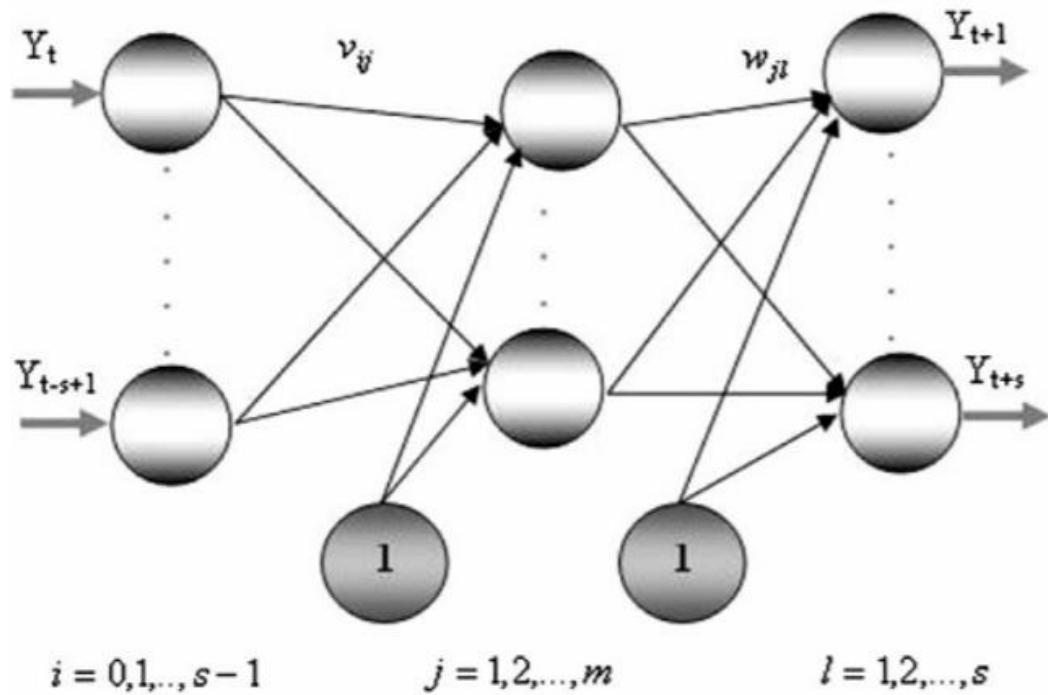


Рисунок 2.4-Архітектура SANN для сезонних часових рядів

Математичним виразом для виведення моделі є:

$$Y_{t+l} = \alpha_l + \sum_{j=1}^m w_{jl} f \left(\theta_j + \sum_{i=0}^{s-1} v_{ij} Y_{t-i} \right) \quad \forall t; l = 1, 2, 3, \dots, s$$

де $Y_{t+l} (l = 1, 2, 3, \dots, s)$ прогнози на майбутні s періодів;
 $Y_{t-i} (i = 1, 2, 3, \dots, s-1)$ спостереження попередніх s періодів;
 $v_{ij} (i = 1, 2, 3, \dots, s-1; j = 1, 2, 3, \dots, m)$ ваги з'єднань від вхідних вузлів до прихованих вузлів;
 $w_{jl} (j = 1, 2, 3, \dots, m; l = 1, 2, 3, \dots, s)$ ваги з'єднань від прихованих вузлів до вихідних вузлів;
 $\alpha_j (j = 0, 1, 2, \dots, q)$ і $\theta_j (j = 0, 1, 2, \dots, m)$ члени зміщення;
 f - активаційна функція.

Таким чином, при прогнозуванні за допомогою SANN кількість нейронів на вході і виході слід брати рівну 12 для місячних і 4 для квартальних

часових рядів. Відповідна кількість прихованих вузлів має визначатися шляхом проведення відповідних експериментів з даними тренування.

2.4 Прогнозні показники ефективності

Після ознайомлення з різними моделями прогнозування часових рядів виникає наступне важливе питання - реалізація, тобто застосування цих методів для складання прогнозів. При застосуванні конкретної моделі до якогось реального чи імітованого часового ряду, спочатку необроблені дані діляться на дві частини, а саме: тренувальний і тестовий набори. Спостереження в навчальному комплекті використовуються для побудови бажаної моделі. Часто невелика частина навчального комплекту зберігається для цілей валідації і відома як валідаційні комплект. Іноді препроцесинг виконується шляхом нормалізації даних, логарифмічних або інших перетворень. Одним з таких відомих методів є метод Box-Cox Transformation. Як тільки модель побудована, вона використовується для створення прогнозів. Спостереження тестового набору проводяться для перевірки точності передбачення цих значень за допомогою відповідної моделі. При необхідності на прогнозовані значення застосовується зворотне перетворення для їх перетворення у вихідну шкалу. Для того, щоб оцінювати точність прогнозування конкретної моделі або для оцінки і порівняння різних моделей розглядається їх відносна продуктивність в наборі тестових даних.

У зв'язку з фундаментальною важливістю прогнозування часових рядів у багатьох практичних ситуаціях, при виборі конкретної моделі слід проявляти належну обережність. З цієї причини, різні показники для оцінки роботи пропонуються в літературі для оцінки прогнозу і порівняти різні моделі. Вони також відомі як показники продуктивності. Кожна з цих перевірок є функцією фактичних і прогнозованих значень часового ряду.

Тепер варто обговорити дані показники ефективності і їх важливі властивості. У кожному з наступних визначень y_t - фактичне значення, f_t – прогнозоване, $e_t = y_t - f_t$ значення похибка прогнозу і n - розмір тестового набору. Також, $\bar{y} = \frac{1}{n} \sum_{t=1}^n y_t$ - середнє значення та $\sigma^2 = \frac{1}{n-1} \sum_{t=1}^n (y_t - \bar{y})^2$ тестова дисперсія.

Помилка середнього прогнозу (The Mean Forecast Error (MFE))

$$MFE = \frac{1}{n} \sum_{t=1}^n e_t$$

Властивості MFE такі:

- Це міра середнього відхилення прогнозних значень від фактичних.
- Він показує напрямок похибки і тому також називається зміщенням прогнозу.
- У MFE вплив позитивних і негативних похибок зводиться нанівець, і немає ніякої можливості точно знати їх кількість.
- Нульовий MFE не означає, що прогнози ідеальні, тобто не містять помилок; скоріше, він тільки вказує на те, що прогнози націлені на потрібну мету.
- MFE не виводить на панель екстремальні помилки.
- Залежить від масштабу виміру, а також залежить від трансформації даних.
- Для хорошого прогнозу, тобто для забезпечення мінімальної похибки, бажано, щоб MFE був якомога ближче до нуля, наскільки це можливо.

Абсолютна помилка (The Mean Absolute Error (MAE))

$$MAE = \frac{1}{n} \sum_{t=1}^n |e_t|$$

Властивості MAE такі:

- Вимірює середнє абсолютне відхилення прогнозованих значень від вихідних.
- Також називається Середнім абсолютним відхиленням (MAD).
- Показує величину загальної помилки, що виникла в результаті прогнозування.
- У MAE вплив позитивних і негативних помилок не зводиться нанівець.
- На відміну від MFE, MAE не дає уявлення про напрямлення помилок.
- Для хорошого прогнозу отримане значення MAE має бути якомога менше.
- Як і MFE, MAE також залежить від масштабу вимірювань і перетворення даних.
- Екстремальні помилки прогнозу не відображаються в панелі MAE.

Середня абсолютна процентна похибка (The Mean Absolute Percentage Error (MAPE))

$$MAPE = \frac{1}{n} \sum_{t=1}^n \left| \frac{e_t}{y_t} \right| \times 100$$

Властивості MAPE такі:

- Цей показник являє собою відсоток від середньої абсолютної похибки, що мала місце.
- Він не залежить від масштабу виміру, але схильний до впливу трансформації даних.
- Він не показує напрямок помилки.
- Панелі MAPE не призначені для панелей з екстремальними відхиленнями.
- При цьому зустрічні знакові похибки не компенсуються одна одною.

Середня відсоткова похибка (The Mean Percentage Error (MPE))

$$MPE = \frac{1}{n} \sum_{t=1}^n \left(\frac{e_t}{y_t} \right) \times 100$$

Властивості MPE такі:

- MPE є відсоток від середньої помилки, що виникла при прогнозуванні.
- Він має ті ж властивості, що й MAPE, за винятком того, що показує напрямки помилки.
- Протилежні помилки впливають один на одного і нівелюють одна одну.
- Таким чином, як і MFE, отримуючи значення MPE близьке до нуля, ми не можемо зробити висновок, що відповідна модель працює дуже добре.
- Бажано, щоб для хорошого прогнозу отриманий MPE був невеликим.

Середньоквадратична похибка (The Mean Squared Error (MSE))

$$MSE = \frac{1}{n} \sum_{t=1}^n e_t^2$$

Властивості MSE такі:

- Це міра середнього квадратного відхилення прогнозованих значень.
- Оскільки тут протилежні знакові помилки не компенсують один одного, MSE дає загальне уявлення про те чи виникли помилки, що виникла під час прогнозування.
- У ньому відображаються екстремальні помилки, що виникають при прогнозуванні.
- MSE підкреслює той факт, що на загальну похибку прогнозування в значній мірі впливають такі фактори як індивідуальні помилки, тобто великі помилки коштують набагато дорожче дрібних помилок.
- MSE не дає уявлення про напрямлення загальної помилки.
- MSE чутливо реагує на зміну масштабу і трансформацію даних.

- Хоча MSE - хороший показник загальної похибки прогнозування, але не такий інтуїтивно зрозумілий і зрозумілий і щоб його було легко інтерпретувати як інші приклади, що обговорювалися раніше.

Сума квадратичної помилки (The Sum of Squared Error (SSE))

$$SSE = \sum_{t=1}^n e_t^2$$

Властивості SSE такі:

- Він вимірює сумарне квадратичне відхилення прогнозованих спостережень від фактичних значень.
- Властивості SSE такі ж, як і у MSE.

Підписана середня квадратична помилка (The Signed Mean Squared Error (SMSE))

$$SMSE = \frac{1}{n} \sum_{t=1}^n \left(\frac{e_t}{|e_t|} \right) e_t^2$$

Властивості SMSE такі:

- Це те ж саме, що і MSE, за винятком того, що тут зберігається вихідний знак для кожної окремої квадратної помилки.
- SMSE систематизує екстремальні помилки, що виникають при прогнозуванні.
- На відміну від MSE, SMSE також показує напрямки загальної помилки.
- При розрахунку SMSE позитивні і негативні помилки компенсують один одного.
- Як і MSE, SMSE також чутливі до змін масштабу і трансформації даних.

Коренева середня квадратична помилка (The Root Mean Squared Error (RMSE))

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{n} \sum_{t=1}^n e_t^2}$$

Властивості RMSE такі:

- RMSE - це не що інше, як квадратний корінь обчисленого MSE.
- Всі властивості MSE зберігаються і для RMSE.

Нормалізована середня квадратична похибка (The Normalized Mean Squared Error (NMSE))

$$NMSE = \frac{MSE}{\sigma^2} = \frac{1}{\sigma^2 n} \sum_{t=1}^n e_t^2$$

Властивості NMSE такі:

- NMSE нормалізує отримане значення MSE після поділу його на дисперсію тесту.
- Це виважена міра похибки і дуже ефективна при оцінці точності прогнозу моделі.
- Чим менше значення NMSE, тим краще прогноз.
- Інші властивості NMSE такі ж, як і у MSE.

U-статистика Тейла (The Theil's U-statistics)

$$U = \frac{\sqrt{\frac{1}{n} \sum_{t=1}^n e_t^2}}{\sqrt{\frac{1}{n} \sum_{t=1}^n f_t^2} \sqrt{\frac{1}{n} \sum_{t=1}^n y_t^2}}$$

- Це нормалізована міра загальної похибки прогнозу.
- $0 \leq U \leq 1$, $U = 0$ середня арифметична ідеально підходить.

- На цей метод впливає зміна масштабу і трансформація даних.
- Для оцінки хорошою точності прогнозу бажано, щоб U-статистика була близька до нуля.

Представлено десять важливих показників для оцінки точності підігнаної моделі. Кожен з цих методів має унікальні властивості, відмінні від інших. В експериментах краще розглядати більш ніж один критерій ефективності. Це допоможе отримати розумні знання про величину, напрямок загальної помилки прогнозу. З цієї причини для часових рядів аналітики зазвичай використовують більше одного виду методів для винесення суджень.

Висновки до розділу 2

До сих пір ми обговорювали три важливі мережеві архітектури, а саме: FNN, TLNN і SANN, які широко використовуються для прогнозування проблем. Деякі інші типи нейронних моделей також запропоновані в літературі, такі як Вероятностная нейронна мережа (PNN) [20] для завдання класифікації і Узагальнена регрессионная нейронна мережа (GRNN) [20] для завдання регресії. Після визначення конкретної мережевої структури наступним найбільш важливим питанням є визначення оптимальних мережевих параметрів. Кількість мережевих параметрів дорівнює загальній кількості зв'язків між нейронами і термінами зміщення [3, 11].

Бажана мережева модель повинна давати відносно невелику помилку не тільки в межах вибірки. (Навчання), але також і за вибічковими (тестовим) даними [20]. З цієї причини величезна обережність проявляється в тому.

необхідне при виборі кількості вхідних і прихованих нейронів. Однак це непросте завдання. оскільки для вибору цих параметрів відсутні теоретичні вказівки, а часто і самі ці параметри

З цією метою проводяться експерименти, такі як перехресна валідація [3, 8]. Інша серйозна проблема полягає в тому, що неадекватне або велика кількість мережевих параметрів може привести до перенавчання даних [2, 11]. Перетренування дає помилково хорошу посадку всередині зразка, що не дає кращого прогнозу. Щоб покарати додавання додаткових параметрів, необхідно виконати наступні дії можуть бути використані критерії порівняння моделей, такі як AIC і BIC [11, 13]. Мережева обрізка [13] і В цьому відношенні вельми популярний також байесовський алгоритм регуляризації MacKay [11, 20]. Підводячи підсумок, можна сказати, що нейромережі - дивно прості, але потужні технології для часу. серійне прогнозування. Вибір відповідних параметрів мережі має вирішальне значення при використанні нейромереж для прогнозування. Крім того, відповідне перетворення або масштабування даних тренування часто виконується в такий спосіб необхідні для досягнення найкращих результатів.

РОЗДІЛ 3

АРХІТЕКТУРА ТА АНАЛІЗ РЕЗУЛЬТАТІВ РОБОТИ

3.1 Короткий огляд

Отримавши достатньо знань про моделювання та прогнозування часових рядів з попередніх глав, можна приступити до їх реалізації на основі практичних наборів даних.

Всі супутні програми написані на мові MATLAB. Для оцінки прогнозних показників різних методів вимірюються показники MAD, MSE, RMSE, MAPE, і розглядається U-статистика Тейла. Для кожного набору даних представлено отримані результати в наступних підрозділах в табличній формі. Також в цьому розділі використано термін прогнозна діаграма для позначення графіка, відображення контрольних (фактичних) і прогнозованих точок даних. На кожній діаграмі прогнозу, твердий і пунктирною лінією представлені, відповідно, фактичні і прогнозовані спостереження.

Експерименти з використанням всіх методів повторюються багаторазово. Також до деяких наборам даних застосовується масштабування і перетворення даних.

3.2 Щоквартальний набір даних по продажах

На рисунку 3.1 представлені квартальні дані по експорту французької фірми за шість років.

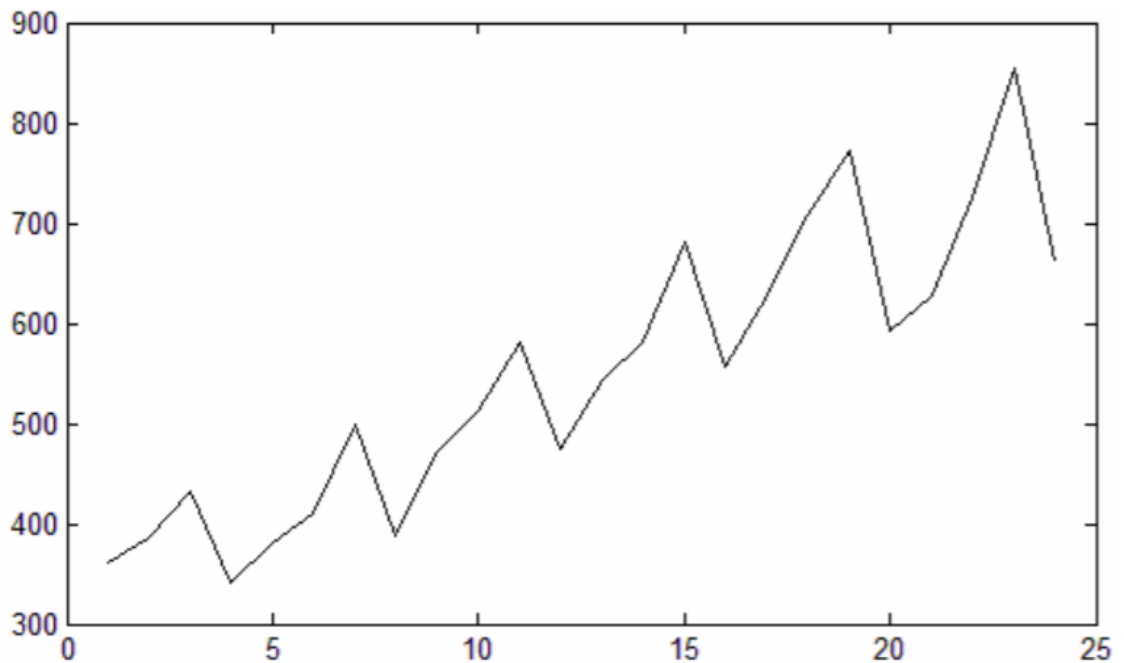


Рисунок 3.1 - Квартальні дані по експорту французької фірми

Це нестационарний сезонний часовий ряд (з сезонним періодом 4), який використовувався К. Хамзачебі для оцінки роботи SANN для квартальних даних. Цей набір даних містить в загальній складності 24 спостереження.

Слідом за К. Хамзачебі мною було збережено перші п'ять років спостережень з щоквартального набору даних з продажу для навчання і шестирічні спостереження для тестування. Після підігнання 4 моделі $SARIMA(0,1,1) \times (0,1,1)^4$ до квартальних рядів продажів після проведення природних логарифмічних перетворень і сезонних відмінностей. Також було встановлено три моделі SANN (варіюючи між собою кількість прихованих вузлів від 1, 2 і до 3). Для монтажу SANNN використано дані про перемасштабування квартальних продажів, отримані після занурення на 100;

Отримані прогностні показники ефективності, розраховані за первісною шкалою за квартал тимчасові ряди продажів представлені в таблиці 3.1:

Таблиця 3.1 прогностні показники ефективності

Метод	MSE	MAD	RMSE	MAPE	Theil's U Statistics
SARIMA(0,1,1) × (0,1,1) ⁴	116.126604	9.740500	10.77621	1.33529%	0.00002
SANN (1 прихований вузол)	968.677397	29.36852	31.12358	4.17214%	0.00006
SANN (2 прихований вузол)	1004.00128	29.48194	31.68598	4.21144%	0.00006
SANN (3 прихований вузол)	466.92096	16.73462	21.60835	2.525232%	0.000041

З таблиці 3.1 видно, що найкращі прогнози квартальних рядів продажів отримані шляхом підгонки 4 моделей SARIMA(0,1,1) × (0,1,1)⁴. Прогностні характеристики моделей SANN є помірними.

На рис. 3.2 представлені діаграми прогнозу продажів SARIMA і SANN (3 прихованих вузла) для квартальних часових рядів продажів:

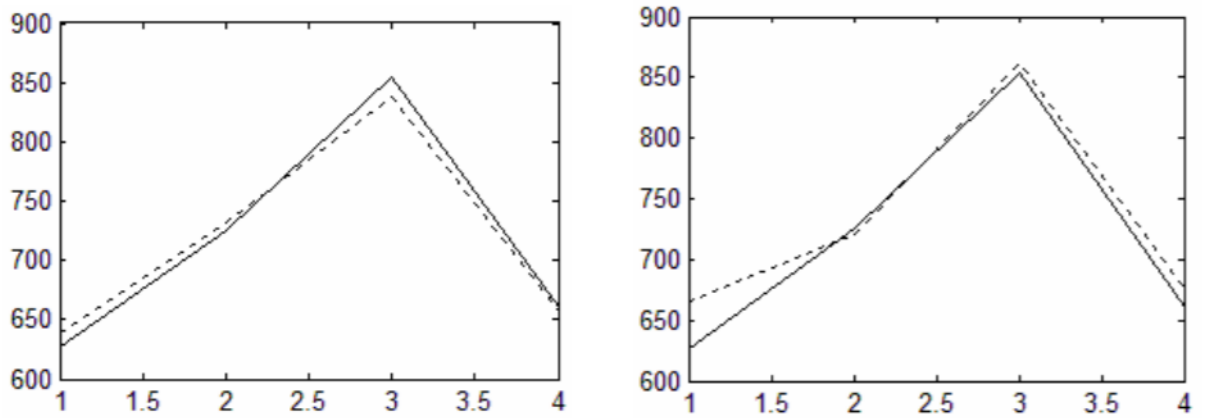


Рисунок 3.2 – Діаграма прогнозів SARIMA і SANN відповідно

З наведених вище двох діаграм прогнозу можна отримати візуальне уявлення про точність прогнозування зазначених моделей SARIMA і SANN для квартальних рядів продажів.

3.3 Щоквартальний набір даних з виробництва в США

На рисунку 3.3 показано щоквартальне виробництво в США в мільйонах з 1975 по 1982 рік.

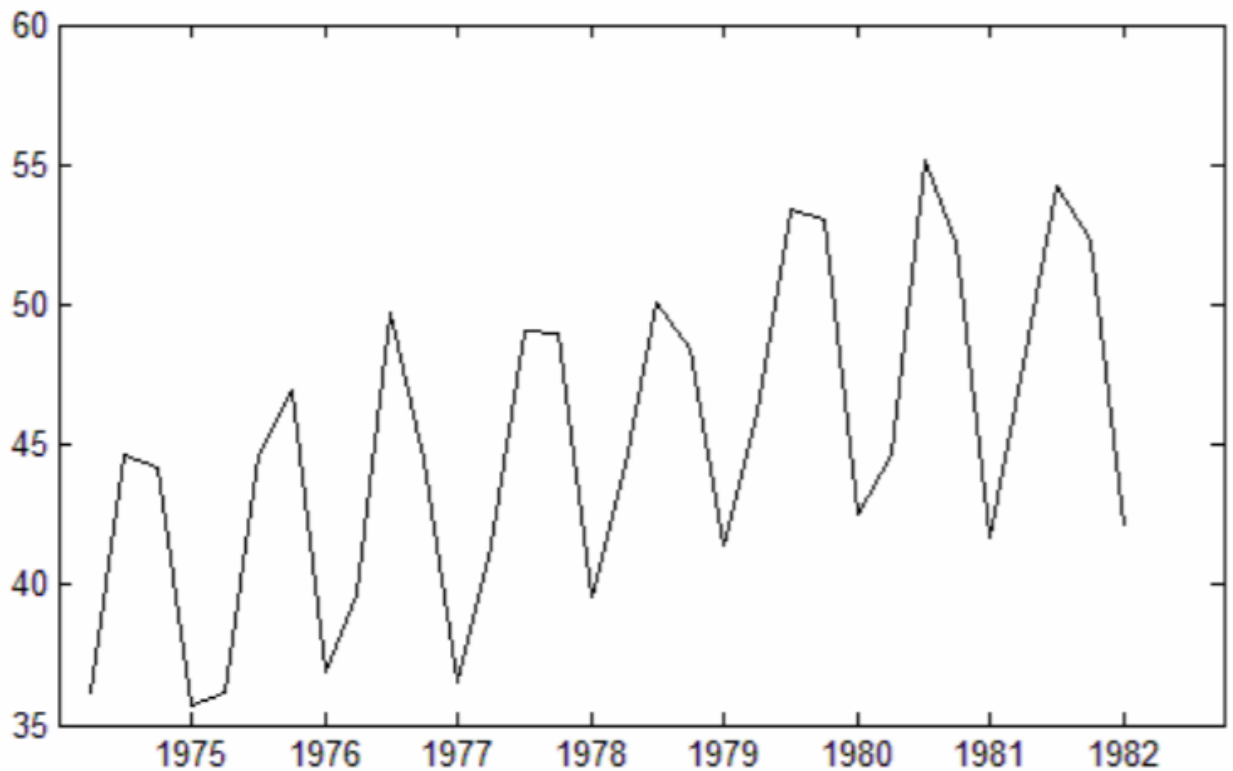


Рисунок 3.3 - Щоквартальне виробництво в США

Цей набір даних часових рядів надано William W. S. Wei; посилання на відповідний веб-сайт приведена в кінці поточної книги. Із загальної кількості 32 спостережень, що містяться в даному часовому ряду, мною використано перші 24 (тобто 1975-1980 рр.) Для підготовки навчальної частини, а решта 8 (тобто 1981-1980 рр.) - для тестової частини. Моделі прогнозування, які було пристосовано до цих часових рядів, такі $SARIMA(0,1,1) \times (0,1,1)^4$, $CSHM$ (з прихованими вузлами 1, 2, 3 і 4) . Як завжди для підгонки моделі $SARIMA$, проведено природне логарифмічна перетворення і сезонні відмінності в значеннях даних. Також для підгонки моделей $SANN$ розділено вихідні повідомлення до 10.

Ми отримали прогнозні показники для щоквартального тимчасового ряду виробництва в США. представлені в вихідній шкалою в таблиці 3.2:

Таблиця 3.2 прогнозні показники ефективності

Метод	MSE	MAD	RMSE	MAPE	Theil's U Statistics
SARIMA(0,1,1) × (0,1,1) ⁴	1.784946	1.195592	1.336019	2.468494%	0.000553
SANN (1 прихований вузол)	2.448568	1.307804	1.564790	2.630288%	0.000066
SANN (2 прихований вузол)	1.868804	1.144692	1.367042	2.364879%	0.000572
SANN (3 прихований вузол)	1.810862	1.076623	1.345683	2.325204%	0.000556
SANN (4 прихований вузол)	1.409691	1.015860	1.187304	2.072790%	0.000496

У таблиці 3.2 показано, що відносно кращі показники ефективності прогнозування для квартальних часових рядів виробництва в США отримані з використанням моделі SANN з чотирма прихованими вузлами. Як видно з таблиці, характеристики моделей SARIMA також досить хороші.

На рисунках 3.3, 3.4 представлені два графіки прогнозу квартальних часових рядів виробництва в США, відповідні моделям SARIMA, SANN (4 прихованих вузла):

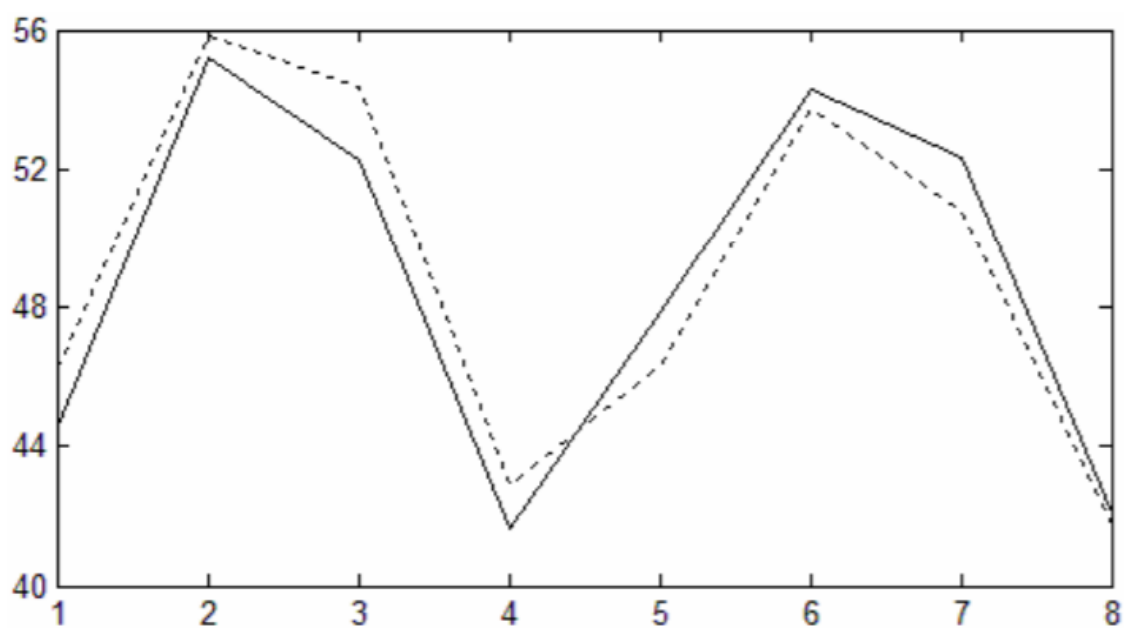


Рисунок 3.3 - Прогноз SARIMA

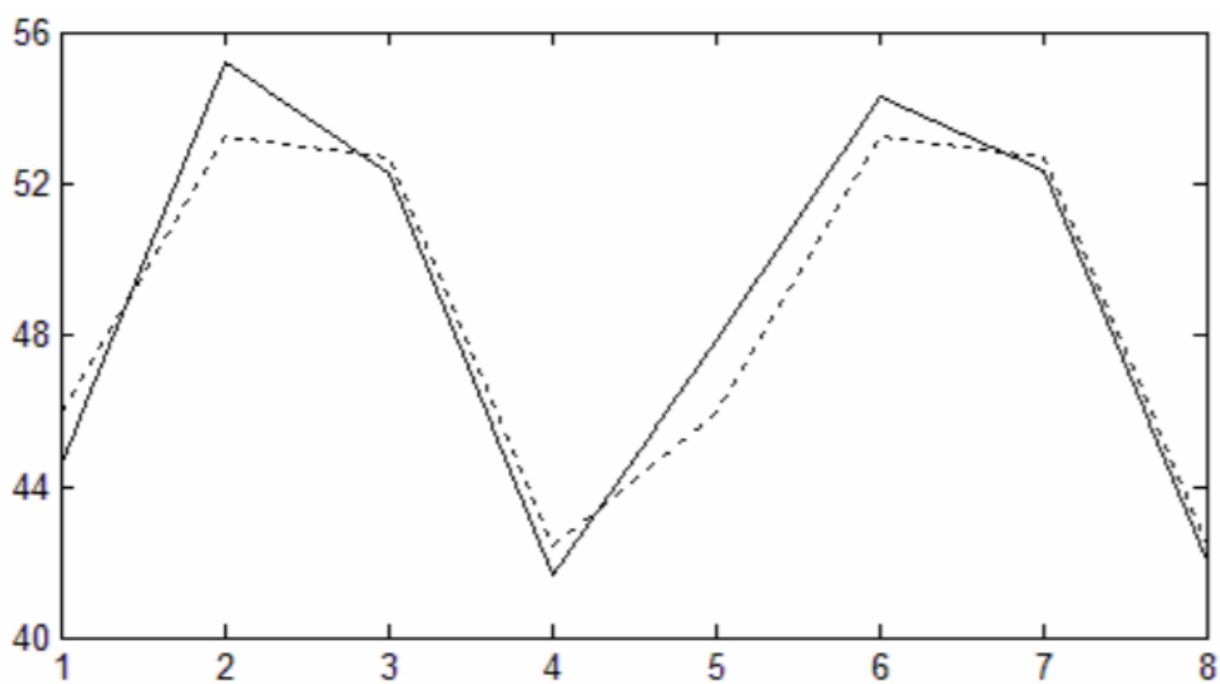


Рисунок 3.4 - Прогноз SANN

Висновки до розділу 3

Багатошарові нейронні мережі - це нейронна мережа прямої дії, яка складається з вхідних шарів вихідних нейронів, по крайній мірі один середній або прихований шар обчислювальних нейронів, нейрони і вихідний шар обчислювальних нейронів. Багатошарові нейронні мережі мають більш ста різних алгоритмів навчання, але найбільш популярним з них є зворотне поширення. Алгоритм зворотнього поширення вибрано для розв'язання задачі оцінки. Порядок моделі ARIMA, тому що це найпопулярніший алгоритм навчання. Він під наглядом відповідну для замовлення моделі ARIMA. Крім того, він дає точні результати результат, оскільки алгоритм зворотнього поширення намагається мінімізувати суму квадратів критерієм помилки, який є корисним індикатором продуктивності мережі, і зближення.

РОЗДІЛ 4

ЕКОНОМІЧНА ЧАСТИНА. ФУНКЦІОНАЛЬНО-ВАРТІСНИЙ АНАЛІЗ

У даному розділі проводиться оцінка основних характеристик програмного продукту, призначеного для прогнозування економічних процесів. Інтерфейс користувача був розроблений за допомогою мови програмування Python у середовищі розробки PyCharm. Інтерфейс користувача створений за допомогою технології Tensor Flow.

Програмний продукт призначено для використання на персональних комп'ютерах під управлінням операційної системи Windows. В даному розділі проводиться аналіз варіантів реалізації модулю з метою вибору оптимальної з економічної точки зору. А саме: проводиться функціонально-вартісний аналіз (ФВА).

Функціонально-вартісний аналіз — це метод комплексного техніко-економічного дослідження об'єкта з метою розвитку його корисних функцій при оптимальному співвідношенні між їхньою значимістю для споживача і витратами на їхнє здійснення. Є одним з основних методів оцінки вартості науково-дослідної роботи, оскільки ФВА враховує як технічну оцінку продукту, що розробляється, так і економічну частину розробки. Крім того, даний метод дозволяє вибрати оптимальний, як з погляду розробника, так і з точки зору покупця варіант розв'язання будь-якої задачі, а також дозволяє оптимізувати витрати й час виконання робіт.

4.1 Постановка задачі техніко-економічного аналізу

У роботі застосовується метод ФВА для проведення техніко-економічний аналізу розробки системи аналізу нелінійних нестационарних процесів. Оскільки основні проектні рішення стосуються всієї системи, кожна окрема підсистема має їм задовольняти. Тому фактичний аналіз представляє собою аналіз функцій програмного продукту, призначеного для збору, обробки та проведення аналізу гетероскедастичних процесів в економіці та фінансах.

Відповідно цьому варто обирати і систему показників якості програмного продукту.

Технічні вимоги до продукту наступні:

- програмний продукт повинен функціонувати на персональних комп'ютерах із стандартним набором компонент;
- забезпечувати високу швидкість обробки великих об'ємів даних у реальному часі;
- забезпечувати зручність і простоту взаємодії з користувачем або з розробником програмного забезпечення у випадку використання його як модуля;
- передбачати мінімальні витрати на впровадження програмного продукту;
- задовольняти умовам допустимих похибок, забезпечувати точність, достатню для прийняття правильних рішень.

4.2 Обґрунтування функцій програмного продукту

Головна функція F_0 – розробка програмного продукту, який аналізує процес за вхідними даними та будує його модель для подальшого прогнозування. Виходячи з конкретної мети, можна виділити наступні основні функції ПП:

F_1 – вибір мови програмування;

F_2 – представлення вихідних даних;

F_3 – функціонал;

F_4 – інтерфейс користувача;

Кожна з основних функцій може мати декілька варіантів реалізації.

Функція F_1 :

а) мова програмування Python;

б) мова програмування R;

Функція F_2 :

а) вивід інформації у файл;

б) вивід інформації на екран;

в) вивід інформації на принтер.

Функція F_3 :

а) мінімальний функціонал, тільки розрахункова частина;

б) ілюстративний функціонал.

Функція F_4 :

а) інтерфейс користувача, створений на базі gWidgets;

б) інтерфейс користувача, створений на базі Tensor Flow.

Морфологічна карта відображає всі можливі комбінації варіантів реалізації функцій, які складають повну множину варіантів програмного продукту і відображена на рисунку 4.1.

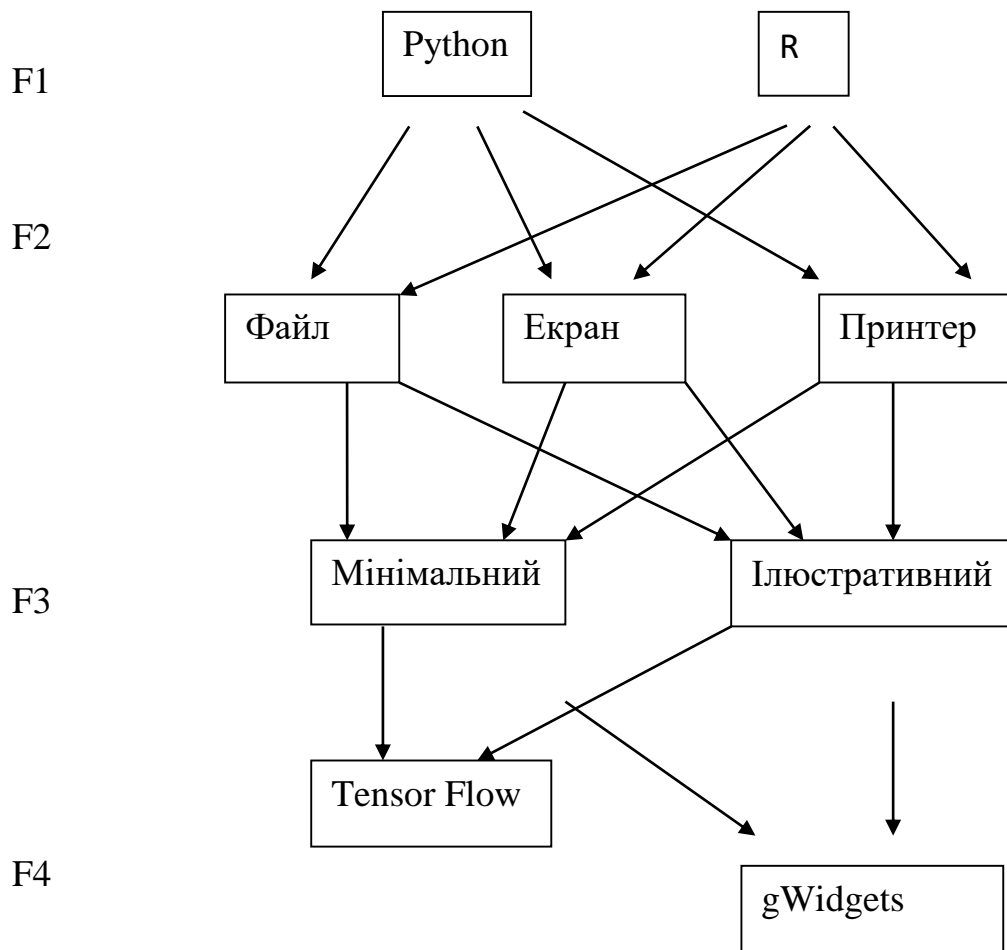


Рисунок 4.1 - Морфологічна карта

На основі цієї карти побудуємо позитивно-негативну матрицю, яка відображена на таблиці 4.1. Вона відображає основні переваги та недоліки кожного з варіантів зазначених функцій.

Таблиця 4.1.Позитивно-негативна матриця

Основна функція	Варіант реалізації	Переваги	Недоліки

F1	а)	Код швидко виконується, поділений на відносно самостійні модулі	Не кросплатформенний, потребує чіткої структуризації ПО
	б)	Код швидко виконується, кросплатформенний	Мало бібліотек і вбудованих функцій
F2	а)	Можливість послідуочого редагування та друку в будь-якому редакторі	Неспівпадіння кодування українських символів в DOS та Windows
	б)	Наочно, не треба шукати файл з результатами	Необхідність перераховувати результати кожен раз навіть для однієї задачі
	в)	Отримання твердої копії результатів	Необхідність наявності принтеру, під'єданого до комп'ютера
F3	а)	Простота реалізації	Складність сприйняття інформації

	б)	Простота сприйняття інформації	Складність реалізації
F4	а)	Легкий у створенні	Відсутність кросплатформеності
	б)	Стабільний у використанні, проста інтеграції з C++	Складний у створенні

На основі аналізу позитивно-негативної матриці робимо висновок, що при розробці програмного продукту деякі варіанти реалізації функцій варто відкинути, тому що вони не відповідають поставленим перед програмним продуктам задачам.

Функція F1: оскільки задача не полягає в тому, щоб описати проблемну область у вигляді сутностей і взаємозв'язків між ними, їх виділення може призвести до ускладнень і неточностей, отже варіант б) потрібно відкинути.

Функція F2: вивід результатів розрахунків на екран не відповідає поставленій задачі, адже ми плануємо користуватись ними довгий час і використовувати надалі для подальших прогнозів; вивід результатів одразу на принтер вимагає, щоб до комп'ютера був під'єднаний кольоровий принтер для коректного відображення графіків, це не зручно, тож залишається варіант а).

Таким чином будемо використовувати наступні варіанти реалізації програмного продукту:

1. F1a – F2a – F3a – F4a ;
2. F1a – F2a – F3a – F4б ;
3. F1a – F2a – F3б – F4a ;

4. F1a – F2a – F3б – F4б .

4.3 Обґрунтування системи параметрів програмного продукту Опис параметрів

Маючи вимоги щодо основних функцій, які реалізуються в програмному продукті, визначають основні параметри виробу, які надалі використовуватимуться для розрахунку коефіцієнта технічного рівня.

Введемо наступні параметри:

- X1 – швидкодія мови програмування;
- X2 – об'єм пам'яті для збереження даних;
- X3 – швидкість обробки та виводу даних;
- X4 – точність розв'язку.

X1: відображає швидкодію операцій залежно від обраної мови програмування

X2: відображає об'єм пам'яті в оперативній пам'яті персонального комп'ютера, необхідний для збереження та обробки даних під час виконання програми.

X3: відображає час, який витрачається на розрахунки;

X4: показує точність розв'язку.

4.3 Кількісна оцінка параметрів

Будемо розглядати 3 типи варіантів значення параметрів, які показані на таблиці 4.2.

Таблиця 4.2. Основні параметри програмного продукту

Назва Параметра	Умовні позначення	Одиниці виміру	Значення параметра		
			гранично допустимі	середні	бажані
Швидкодія мови програмування	X1	Оп/мс	2000	11000	19000
Об'єм пам'яті для збереження даних	X2	Мб	64	32	16
Час розрахунків	X3	мс	800	420	260
Точність розв'язку	X4	%	10	5	1

Бальна оцінка швидкодії мови програмування представлена на рисунку 4.2.

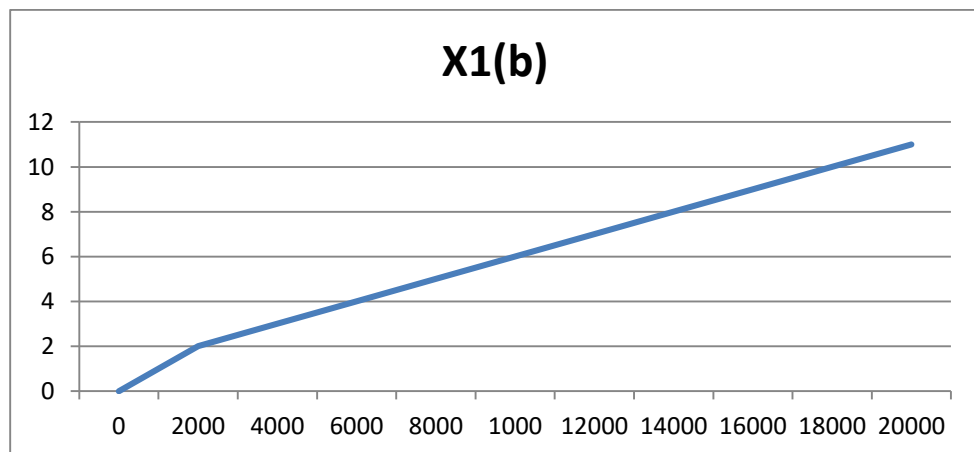


Рисунок 4.2 - Бальна оцінка швидкодії мови програмування

Бальна оцінка об'єму пам'яті для збереження даних представлена на рисунку 4.3.

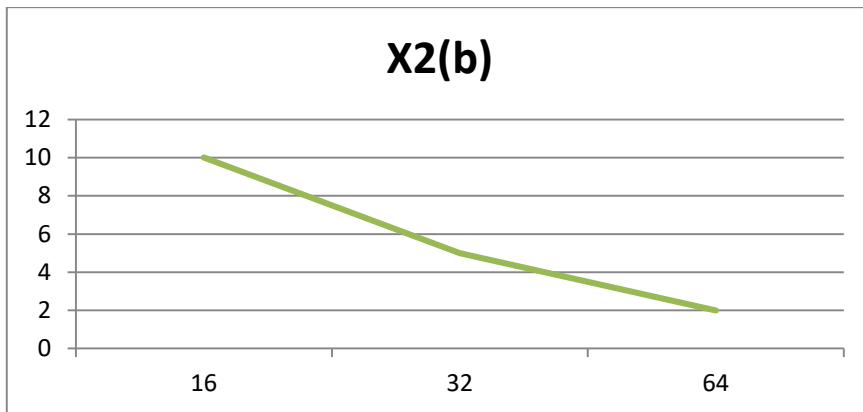


Рисунок 4.3 - Бальна оцінка об'єму пам'яті

Бальна оцінка часу розрахунків представлена на рисунку 4.4.

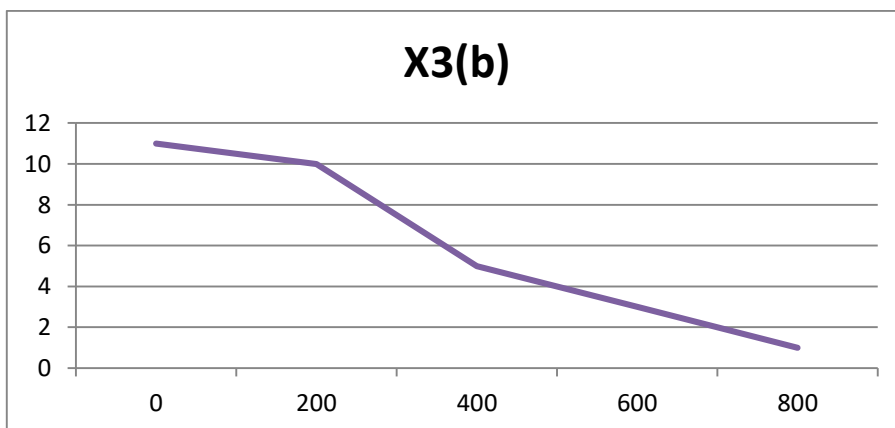


Рисунок 4.4 - Бальна оцінка часу розрахунків

4.3.3 Аналіз експертного оцінювання параметрів

Кожен експерт оцінює ступінь важливості кожного параметру для конкретно поставленої цілі – розробка програмного продукту, який дає

найбільш точні результати при знаходженні апроксимуючого багаточлену. Розрахуємо коефіцієнт конкордації (узгодженості) експертних оцінок. Результати ранжування параметрів відповідно представлені на таблиці 4.3, та результати ранжування показників відповідно продемонстровані на таблиці 4.4.

Таблиця 4.3. Результати ранжування параметрів

Параметри	Експерти									Підсумкова	Числове
	1	2	3	4	5	6	7	8	9	оцінка	значення
X1,X2	<	<	<	<	<	<	>	<	>	<	0,5
X1,X3	<	>	<	<	<	<	>	<	<	<	1,5
X1,X4	<	<	<	<	<	<	<	<	<	<	1,5
X2,X3	>	>	>	<	<	<	>	<	<	<	1,5
X2,X4	<	>	<	<	<	<	<	<	<	<	1,5
X3,X4	<	>	<	>	<	<	<	<	>	<	0,5

Таблиця 4.4 Результати ранжування показників

Познач. параметра	Ранг параметра за оцінкою експерта									Сума рангів R_i	Відхилення Δ_i	Δ_i^2
	1	2	3	4	5	6	7	8	9			
X1	1	2	1	1	1	1	3	1	2	13	-9,5	90,25
X2	3	4	3	2	2	2	1	2	1	20	-2,5	6,25
X3	2	1	2	4	3	3	2	3	4	24	1,5	2,25
X4	4	3	4	3	4	4	4	4	3	33	10,5	110,25
Разом	10	10	10	10	10	10	10	10	10	90	0	209

Порахуємо коефіцієнт узгодженості:

Для перевірки степені достовірності експертних оцінок, визначимо наступні параметри:

а) сума рангів кожного з параметрів і загальна сума рангів:

$$R_i = \sum_j^N r_{ij} R_{ij} = \frac{Nn(n+1)}{2} = 90$$

де N – кількість експертів, n – кількість параметрів;

б) середня сума рангів:

$$T = \frac{1}{n} R_{ij} = 22,5$$

в) відхилення суми рангів кожного параметра від середньої суми рангів:

$$\Delta_i = R_i - T$$

Сума відхилень по всіх параметрам повинна дорівнювати 0;

г) загальна сума квадратів відхилення:

$$S = \sum_{i=1}^n \Delta_i^2 = 209$$

$$W = \frac{12S}{N^2(n^3 - n)}, W = \frac{12 * 209}{(9^2 * (4^3 - 4))} = 0,69 > W_k = 0,67$$

Ранжирування можна вважати достовірним, тому що знайдений коефіцієнт узгодженості перевищує нормативний, котрий дорівнює 0,67. Скориставшись результатами ранжирування, проведемо попарне порівняння

всіх параметрів і результати занесемо у таблицю. Числове значення, що визначає ступінь переваги i -го параметра над j -тим, a_{ij} визначається по формулі:

$$a_{ij} = \begin{cases} 1.5, & X_i > X_j \\ 1, & X_i = X_j \\ 0.5, & X_i < X_j \end{cases}$$

З отриманих числових оцінок переваги складемо матрицю $A = \| a_{ij} \|$.

Для кожного параметра зробимо розрахунок вагомості K_{Bi} за наступними формулами:

$$\hat{E}_{\hat{a}^3} = \frac{b_i}{\sum_{i=1}^n b_i}, \text{ де } b_i = \sum_{i=1}^n a_{ij}$$

Відносні оцінки розраховуються декілька разів доти, поки наступні значення не будуть незначно відрізнятися від попередніх (менше 2%). На другому і наступних кроках відносні оцінки розраховуються за наступними формулами:

$$\hat{E}_{\hat{a}^3} = \frac{b'_i}{\sum_{i=1}^n b'_i}, \text{ де } b'_i = \sum_{j=1}^n a_{ij} b_j$$

Таблиця 5. Розрахунок вагомості параметрів

Параметри	Параметри				Перший крок		Другий крок		Третій крок	
	X1	X2	X3	X4	b_i	K_{Bi}	b_i^1	$\hat{E}_{\hat{a}^3}^1$	b_i^2	$\hat{E}_{\hat{a}^3}^2$

X1	1,0	0,5	0,5	0,5	2,5	0,17	9,25	0,157	34,125	0,159
X2	1,5	1,0	0,5	0,5	3,5	0,23	12,25	0,208	44,65	0,208
X3	1,5	1,5	1,0	0,5	4,5	0,30	16,25	0,274	58,45	0,272
X4	1,5	1,5	1,5	1,0	5,5	0,37	21,25	0,360	77,875	0,362
Всього:					15	1	59	1	215	1

Як видно з таблиці, різниця значень коефіцієнтів вагомості не перевищує 2%, тому більшої кількості ітерацій не потрібно.

4.3.4 Аналіз рівня якості варіантів реалізації функцій

Визначаємо рівень якості кожного варіанту виконання основних функцій окремо.

Абсолютні значення параметрів X2(об'єм пам'яті для збереження даних) та X1(швидкодія мови програмування) відповідають технічним вимогам умов функціонування даного ПП.

Абсолютне значення параметра X3 (час розрахунків) обрано не найгіршим (не максимальним), тобто це значення відповідає або варіанту б) 420 мс або варіанту в) 260мс.

Абсолютні значення параметра X4 (точність розв'язку) обрані з наступних міркувань: для варіанта б) X4=5%, тому що такий варіант підходить для клієнтів не дуже вимогливих до точності розв'язку; для варіанта в) X4=1% – висока точність розв'язку є необхідною умовою.

Коефіцієнт технічного рівня для кожного варіанта реалізації ПП розраховується за формулою:

$$K_{TP} = \sum_i K_{ei} \cdot B_i, i = 1..n,$$

де n – кількість параметрів, Kv_i – коефіцієнт вагомості i -го параметра, B_i – оцінка i -го параметра в балах. Розрахунок показників рівня якості представлено відповідно на таблиці 4.6.

Таблиця 4.6. Розрахунок показників рівня якості варіантів реалізації

Основні функції	Варіант реалізації функції	Абсолютне значення параметра	Бальна оцінка параметра	Коефіцієнт вагомості параметра	Коефіцієнт рівня якості
F1	а	11000	5	0,159	0,795
F2	а	32	5	0,208	1,04
F3	а	420	5	0,272	1,36
	б	260	10	0,272	2,72
F4	а	5	5	0,362	1,81
	б	1	10	0,362	3,62

За цими даними визначаємо рівень якості кожного з варіантів:

$$K_{TEP1} = 0,795 + 1,04 + 1,36 + 1,81 = 5,005$$

$$K_{TEP2} = 0,795 + 1,04 + 1,36 + 3,62 = 6,815$$

$$K_{TEP3} = 0,795 + 1,04 + 2,72 + 1,81 = 5,365$$

$$K_{TEP4} = 0,795 + 1,04 + 2,72 + 3,62 = 7,175$$

Отже, найкращим є четвертий варіант, для якого коефіцієнт технічного рівня має найбільше значення.

4.4 Економічний аналіз варіантів розробки ПП

Для визначення вартості розробки ПП спочатку проведемо розрахунок трудомісткості. Варіанти включають в себе два окремих завдання:

1. Розробка проекту програмного продукту;

2. Розробка програмної оболонки.

Завдання 1 за ступенем новизни відноситься до групи А, завдання 2 – до групи Б. За складністю алгоритми, які використовуються в завданні 1 належать до групи 1; а в завданні 2 – до групи 3.

Для реалізації завдання 1 використовується довідкова інформація, а завдання 2 використовує інформацію у вигляді даних.

Проведемо розрахунок норм часу на розробку та програмування для кожного з завдань.

Проведемо розрахунок норм часу на розробку та програмування для кожного з завдань. Загальна трудомісткість обчислюється як

$$T_O = T_P \cdot K_P \cdot K_{СК} \cdot K_M \cdot K_{СТ} \cdot K_{СТ.М}, \quad (4.1)$$

де T_P – трудомісткість розробки ПП; K_P – поправочний коефіцієнт; $K_{СК}$ – коефіцієнт на складність вхідної інформації; K_M – коефіцієнт рівня мови програмування; $K_{СТ}$ – коефіцієнт використання стандартних модулів і прикладних програм; $K_{СТ.М}$ – коефіцієнт стандартного математичного забезпечення

Для першого завдання, виходячи із норм часу для завдань розрахункового характеру степеню новизни А та групи складності алгоритму 1, трудомісткість дорівнює: $T_P = 90$ людино-днів. Поправочний коефіцієнт, який враховує вид нормативно-довідкової інформації для першого завдання: $K_P = 1.7$. Поправочний коефіцієнт, який враховує складність контролю вхідної та вихідної інформації рівний 1: $K_{СК} = 1$. Оскільки при розробці першого завдання використовуються стандартні модулі, врахуємо це за допомогою коефіцієнта $K_{СТ} = 0.8$. Тоді, за формулою 5.1, загальна трудомісткість програмування першого завдання дорівнює:

$$T_1 = 90 \cdot 1.7 \cdot 0.8 = 122.4 \text{ людино-днів.}$$

Проведемо аналогічні розрахунки для подальших завдань.

Для другого завдання (використовується алгоритм третьої групи складності, степінь новизни Б), тобто $T_P = 27$ людино-днів, $K_{II} = 0.9$, $K_{СК} = 1$, $K_{СТ} = 0.8$:

$$T_2 = 27 \cdot 0.9 \cdot 0.8 = 19.44 \text{ людино-днів.}$$

Складаємо трудомісткість відповідних завдань для кожного з обраних варіантів реалізації програми, щоб отримати їх трудомісткість:

$$T_I = (122.4 + 19.44 + 4.8 + 19.44) \cdot 8 = 1328.64 \text{ людино-годин;}$$

$$T_{II} = (122.4 + 19.44 + 6.91 + 19.44) \cdot 8 = 1345.52 \text{ людино-годин;}$$

$$T_{III} = (122.4 + 19.44 + 6.8 + 19.44) \cdot 8 = 1344.64 \text{ людино-годин;}$$

$$T_{IV} = (122.4 + 19.44 + 6 + 19.44) \cdot 8 = 1338.24 \text{ людино-годин;}$$

Найбільш високу трудомісткість має варіант II.

В розробці беруть участь два програмісти з окладом 6000 грн., один фінансовий аналітик з окладом 9000 грн. Визначимо зарплату за годину за формулою:

$$C_q = \frac{M}{T_m \cdot t} \text{ грн.,}$$

де M – місячний оклад працівників; T_m – кількість робочих днів тиждень; t – кількість робочих годин в день.

$$C_q = \frac{6000 + 6000 + 9000}{3 \cdot 21 \cdot 8} = 41.67 \text{ грн.}$$

Тоді, розрахуємо заробітну плату за формулою

$$C_{зп} = C_q \cdot T_i \cdot K_d,$$

де C_q – величина погодинної оплати праці програміста;

T_i – трудомісткість відповідного завдання;

K_d – норматив, який враховує додаткову заробітну плату.

Зарплата розробників за варіантами становить:

$$I. \quad C_{зп} = 41,67 \cdot 1328.64 \cdot 1.2 = 66437,31 \text{ грн.}$$

$$II. \quad C_{зп} = 41,67 \cdot 1345.52 \cdot 1.2 = 67281,38 \text{ грн.}$$

$$III. \quad C_{зп} = 41,67 \cdot 1344.64 \cdot 1.2 = 67237,38 \text{ грн}$$

$$IV. \quad C_{зп} = 41,67 \cdot 1338.24 \cdot 1.2 = 66917,35 \text{ грн}$$

Відрахування на єдиний соціальний внесок в залежності від групи професійного ризику (II клас) становить 22%

$$I. \quad C_{від} = C_{зп} \cdot 0.3677 = 66437,31 \cdot 0.22 = 14616,21 \text{ грн.}$$

$$II. \quad C_{від} = C_{зп} \cdot 0.3677 = 67281,38 \cdot 0.22 = 14801,90 \text{ грн.}$$

$$III. \quad C_{від} = C_{зп} \cdot 0.3677 = 67237,38 \cdot 0.22 = 14792,22 \text{ грн.}$$

$$IV. \quad C_{від} = C_{зп} \cdot 0.3677 = 66917,35 \cdot 0.22 = 14721,82 \text{ грн.}$$

Тепер визначимо витрати на оплату однієї машино-години. (C_m)

Оскільки, одна ЕОМ обслуговує одного програміста з окладом 6000 грн., з коефіцієнтом зайнятості 0,2 то для однієї машини отримаємо:

$$C_{г} = 12 \cdot M \cdot K_3 = 12 \cdot 6000 \cdot 0,2 = 14400 \text{ грн.}$$

З урахуванням додаткової заробітної плати:

$$C_{зп} = C_{г} \cdot (1 + K_3) = 14400 \cdot (1 + 0.2) = 17280 \text{ грн.}$$

Відрахування на єдиний соціальний внесок:

$$C_{від} = C_{зп} \cdot 0.22 = 17280 \cdot 0,22 = 3801,60 \text{ грн.}$$

Амортизаційні відрахування розраховуємо при амортизації 25% та вартості ЕОМ – 8000 грн.

$$C_A = K_{тм} \cdot K_A \cdot Ц_{пр} = 1.15 \cdot 0.25 \cdot 8000 = 2300 \text{ грн.,}$$

де $K_{тм}$ – коефіцієнт, який враховує витрати на транспортування та монтаж приладу у користувача; K_A – річна норма амортизації; $Ц_{пр}$ – договірна ціна приладу.

Витрати на ремонт та профілактику розраховуємо як:

$$C_P = K_{тм} \cdot Ц_{пр} \cdot K_P = 1.15 \cdot 8000 \cdot 0.05 = 460 \text{ грн.,}$$

де K_P – відсоток витрат на поточні ремонти.

Ефективний годинний фонд часу ПК за рік розраховуємо за формулою:

$$T_{\text{ЕФ}} = (D_K - D_B - D_C - D_P) \cdot t_3 \cdot K_B = (365 - 104 - 8 - 16) \cdot 8 \cdot 0.9 = 1706.4$$

годин,

де D_K – календарна кількість днів у році; D_B , D_C – відповідно кількість вихідних та святкових днів; D_P – кількість днів планових ремонтів устаткування; t – кількість робочих годин в день; K_B – коефіцієнт використання приладу у часі протягом зміни.

Витрати на оплату електроенергії розраховуємо за формулою:

$$C_{\text{ЕЛ}} = T_{\text{ЕФ}} \cdot N_C \cdot K_3 \cdot C_{\text{ЕН}} = 1706,4 \cdot 0,156 \cdot 2,7515 \cdot 2 = 1464,89 \text{ грн.},$$

де N_C – середньо-споживча потужність приладу; K_3 – коефіцієнтом зайнятості приладу; $C_{\text{ЕН}}$ – тариф за 1 КВт-годин електроенергії

Накладні витрати розраховуємо за формулою:

$$C_H = C_{\text{ПР}} \cdot 0.67 = 8000 \cdot 0,67 = 5360 \text{ грн.}$$

Тоді, річні експлуатаційні витрати будуть:

$$C_{\text{ЕКС}} = C_{\text{ЗП}} + C_{\text{ВІД}} + C_A + C_P + C_{\text{ЕЛ}} + C_H$$

$$C_{\text{ЕКС}} = 17280 + 3801,60 + 2300 + 460 + 1464,89 + 5360 = 30666,49 \text{ грн.}$$

Собівартість однієї машино-години ЕОМ дорівнюватиме:

$$C_{\text{М-Г}} = C_{\text{ЕКС}} / T_{\text{ЕФ}} = 30666,49 / 1706,4 = 17,97 \text{ грн/час.}$$

Оскільки в даному випадку всі роботи, які пов'язані з розробкою програмного продукту ведуться на ЕОМ, витрати на оплату машинного часу, в залежності від обраного варіанта реалізації, складає:

$$C_M = C_{\text{М-Г}} \cdot T$$

$$\text{I. } C_M = 17,97 \cdot 1328,64 = 24819 \text{ грн.};$$

$$\text{II. } C_M = 17,97 \cdot 1345,52 = 25134,52 \text{ грн.};$$

$$\text{III. } C_M = 17,97 \cdot 1344,64 = 24163,18 \text{ грн.};$$

$$\text{IV. } C_M = 17,97 \cdot 1338,24 = 24048,17 \text{ грн.};$$

Накладні витрати складають 67% від заробітної плати:

$$C_H = C_{3П} \cdot 0,67$$

- I. $C_H = 66437,31 \cdot 0,67 = 44513 \text{ грн.};$
- II. $C_H = 67281,38 \cdot 0,67 = 45078,52 \text{ грн.};$
- III. $C_H = 67237,38 \cdot 0,67 = 45049,05 \text{ грн.};$
- IV. $C_H = 66917,35 \cdot 0,67 = 44834,62 \text{ грн.};$

Отже, вартість розробки ПП за варіантами становить:

$$C_{ПП} = C_{3П} + C_{ВІД} + C_M + C_H$$

- I. $C_{ПП} = 66437,31 + 14616,21 + 24819 + 44513 = 150385,52 \text{ грн.};$
- II. $C_{ПП} = 67281,38 + 14801,90 + 25134,52 + 45678,52 = 152696,32 \text{ грн.};$
- III. $C_{ПП} = 67237,38 + 14792,22 + 24163,18 + 45049,05 = 151241,83 \text{ грн.};$
- IV. $C_{ПП} = 66917,35 + 14721,82 + 24048,17 + 44834,62 = 150521,96 \text{ грн.};$

4.5 Вибір кращого варіанта ПП техніко-економічного рівня

Розрахуємо коефіцієнт техніко-економічного рівня за формулою:

$$K_{TEPj} = K_{Kj} / C_{Фj},$$

$$K_{TEP1} = 5,005 / 150385,52 = 0,33 \cdot 10^{-4};$$

$$K_{TEP2} = 6,815 / 152696,32 = 0,46 \cdot 10^{-4};$$

$$K_{TEP3} = 5,365 / 151241,83 = 0,35 \cdot 10^{-4};$$

$$K_{TEP4} = 7,175 / 150521,96 = 0,47 \cdot 10^{-4};$$

Як бачимо, найбільш ефективним є четвертий варіант реалізації програми з коефіцієнтом техніко-економічного рівня $K_{TEP4} = 0,47 \cdot 10^{-4}$.

Висновки до розділу 4

В результаті виконання економічного розділу були систематизовані і закріплені теоретичні знання в галузі економіки та організації виробництва використанням їх для техніко-економічного обґрунтування розробки методом функціонально-вартісного аналізу.

На основі даних про зміст основних функцій, які повинен реалізувати програмний продукт, були визначені чотири найбільш перспективні варіанти реалізації продукту. Найбільш ефективним виявився четвертий варіант реалізації функцій ПП, який дає максимальну величину коефіцієнта техніко-економічного рівня $K_{\text{ТЕР}}$, вартість витрат для нього становить $C_{\text{ПП}} = 150521,96$ грн.

Цей варіант передбачає:

- мова програмування Python;
- зберігання вихідних даних у файл;
- використання ілюстративних матеріалів;
- інтерфейс користувача на базі Tensor Flow;
- найвищу точність розрахунків.

ВИСНОВКИ ПО РОБОТІ ТА ПЕРСПЕКТИВИ ПОДАЛЬШИХ ДОСЛІДЖЕНЬ

У загальному, представлено сучасний стан справ в області слідуючи популярним моделям прогнозування часових рядів з їх характерними особливостями:

- Box-Jenkins або ARIMA моделі для прогнозування лінійних часових рядів.
- Деякі нелінійні стохастичні моделі, такі як NMA, ARCH.
- Моделі прогнозування нейромереж; TLNNN і SANN.

Було відмічено, що при правильному підборі моделі (в разі ARIMA), число вхідних, прихованих і вихідних нейронів (в разі ШНМ) надзвичайно важливий для успішного прогнозування. Було описано важливі функції, такі як AIC і BIC, які часто використовуються для вибору моделі ARIMA.

Для вибору кількості відповідних нейронів в ШНМ перехресна валідація повинна бути проведена, як згадувалося вище.

Розглянуто кілька важливих показників оцінки роботи для оцінки точності моделі прогнозування. Зрозуміло, що для отримання адекватних результатів про похибку прогнозування, на практиці слід використовувати більше одного виду критеріїв перевірок.

Остання глава містить прогнозні результати експериментів, виконаних по двом наборах даних в режимі реального часу. Розуміння розглянутих моделей прогнозування та їх успішності процесу виконання можна спостерігати за п'ятьма показниками оцінки роботи і прогнозом діаграми, які отримані для кожного з двох наборів даних. Проте, в деяких випадках, можна побачити відхилення між первинними спостереженнями і прогнозними значеннями. У таких випадках, ми можемо припустити, що відповідна попередня обробка даних, відмінна від тієї, яка використана в моїй роботі може поліпшити прогнозні показники.

Прогнозування часових рядів швидко розвивається і надає широкі можливості для майбутньої роботи. Одним з них є Комбінований підхід, тобто поєднання декількох різних підходів і різні методи підвищення точності прогнозів.

Разом з іншими аналітичними дослідженнями в галузі прогнозування часових рядів, ми подумали про те, щоб знайти ефективний спосіб комбінуючи модель, в майбутньому, якщо це можливо